Improving Mobile Video

Alexandre Chapiro
Advisor: Paulo Cezar Pinto Carvalho
Co-Advisor: Luiz Velho
Visgraf
Instituto de Matemática Pura e Aplicada

A thesis submitted for the degree of

Master of Science in Mathematics (Major - Computer Graphics)

Date: 23rd of August - 2011

1. Reviewer:

2. Reviewer:

3. Reviewer:

Day of the defense:

Signature from head of PhD committee:

Abstract

This dissertation deals with a well known problem - the fact that consumer cameras are unable to capture the whole range of color luminance variations the human eye is able to perceive. Many widespread techniques, especially those related to High Dynamic Range, deal with this issue, focusing mainly on still images. Our focus in this work, however, is to study possibilities of the improvement of videos taken from mobile devices by the consumer through the application or extension of these techniques.

Acknowledgements

"Why is this thus? What is the reason for this thusness?"
Artemus Ward

The quotation above encompasses what probably is the central motif of Science. For teaching me these and other lessons, I would like to thank my parents, who have always cared for me and made the greatest efforts to bring me the best opportunities, professionally and otherwise.

I would like to thank my advisors, Paulo Cezar Pinto Carvalho and Luiz Velho for their mentoring and valuable advice, without which this work would have been impossible. To all my teachers at the Institute of Pure and Applied Mathematics - thank you very much.

I would also like to thank all my colleagues from the Visgraf laboratory, which is a wonderful environment for learning.

Finally, I would like to thank all those who were close to me during my Master's at IMPA for their support and friendship, without which life would have been quite dull.

Contents

List of Figures **v**

1 Introduction **1**

1.1 Historical Background 1

2 High Dynamic Range—an Overview **7**

2.1 HDR—an overview 7

2.1.1 Recovering the Camera Response Function 7

2.2 Creating HDR images 9

3 Towards Mobile HDR **13**

3.1 Introduction 13

3.2 Previous Work 14

3.3 HDR Video 15

3.3.1 Photometric Calibration 16

3.3.2 Histogram-Based Registration 16

3.3.3 Radiance Map Reconstruction 17

3.4 Mobile Device Implementation 17

3.4.1 Capturing HDR Video 18

3.4.2 Processing HDR video 18

3.5 Results and Discussion 19

3.6 Conclusions and Future Work 19

4 Alternative Methods - Exposure Fusion **23**

4.1 Exposure Fusion Pixel Weighting System 23

4.2 Additional Considerations 25

5 Exposure Fusion Video **31**

5.1 First Attempts 31

5.2 Deghosting Method 34

5.2.1 Acquisition and Registration of Input Images 36

5.3 Results and Future Work 36

6 Conclusions **39**

References **41**

List of Figures

1.1 In the image above, on the left - chemical photographic film is shown. On the right is a more modern electronic sensor, as present in digital cameras. 2

1.2 A picture showing how the camera's limited dynamic range can sometimes lead to loss of important information on the scene. In this case, the guitar player is completely obscured due to the camera adjusting to a strong background lighting (taken by Luciano Harper). 3

1.3 An example of the creation of an HDR image from several differently exposed LDR images. The HDR image must then be tone-mapped for a viewable result. Images from Photomatrix. 4

1.4 This image shows the result of using Exposure Fusion to combine several LDR images into another, albeit more pleasant, LDR image. (Image by Seb Przd). 5

2.1 This image shows the response curve of a Nokia N900 cellphone. The horizontal axis represents a log-scale of radiosity values, the vertical scale represents pixel values. The bell curve shows the weights given to each pixel value during the process, with lesser weights being given to pixels with poor exposures (the ones closer to minimum and maximum values). This last step is explained below. 9

2.2 This image shows a log-scale axis with Illuminance values, measured in $Candelas/Meter^2$. 10

2.3 A picture showing an example of HDR imaging. Note that none of the input images contain all the details present in the final image, even though it has been tone-mapped to fit the display device's limited dynamic range. Image borrowed from pfstools. 12

3.1 (Top) Captured frames. From left to right: optimally-, sub- and super-exposed shots. (Bottom) Corresponding results with tone-mapping. . . . 20

3.2 (Top) Captured frames. From left to right: sub-, normal- and super-exposed shots, according to initial calibration on a brighter scene. (Bottom) Corresponding results with tone-mapping. 20

3.3 Another set of results. The HDR image is presented in a red frame. Notice the details in the sky, clothes and vegetation, missing from the original images. 21

4.1 A comparisson between tone-mapped HDRI (left) and Exposure Fusion (right) created using commercial software, taken from Digital Photography School. 24

4.2 (Top Left) Four images with varying exposures used as input. (Top Right) Resulting Exposure-Fused image. (Center Left) Final weight map for the third image in the stack, obtained from the product of the three weight maps below. (Center Right) Weight map for the Contrast parameter for the third image in the stack. (Bottom Left) Same for the Saturation parameter. (Bottom Right) Same for the Well-exposedness parameter. 27

4.3 An example of a regular photograph (left) fused with a photograph taken with flash (right). Notice how the over-exposed glare on the painting's glass is removed by the algorithm using the Well-exposedness parameter. Image borrowed from (1). 28

4.4 An example of ghosting caused by object movement on the scene. The moving man in the input images (left) is averaged through the fusion process and has a transparent appearance in the final image (right). . . 29

5.1 Simple calculation of absolute difference between pixel regions. 33

5.2 Previously seen coefficients after processing. 34

5.3 A favorable result of the algorithm. 34
5.4 Input images shown on the left side, the (amplified x5) result of a High-Pass Laplacian Filter on the right. 35
5.5 On the left - resuls of the local analysis for several Low-Pass applications. On the right, the final deghosting parameter. 36
5.6 Video Creation Pipeline: (top-left) Inputs, (top-right) Ghosting Coefficients after High-Pass filtering (bottom-right) Deghosted result (bottom-left) Comparison. 37

1

Introduction

1.1 Historical Background

A Dictionary Definition:

Photography

noun

- the art or process of producing images by the action of radiant energy and especially light on a sensitive surface.

The drive to create depictions resembling the world we live in has been present in human minds from ancient history. It was more recently, however, that the creation of mobile devices that allow us to use light in order to create images resembling the real world came to be. Cameras have now been a part of human daily lives for at least a century and through this time they have endured many changes. One such change laid in the replacement of chemical films as a medium to measure incoming light for digital censors, thus giving birth to what we call "digital cameras" (see Fig. 1.1).

Several advantages in the use of digital cameras have brought them to a dominance in the photography scenario for all but a few niche markets. Some of these include the ease with which device memory is used to store digital images, as in contrast to previous film technology. In fact, digital photography allows the users of modern digital cameras to take a very large number of photographs without need for care of a physical medium or post-processing. Although these circumstances, coupled with the explosive growth of digital technologies in the past decades, have only served to consolidate the

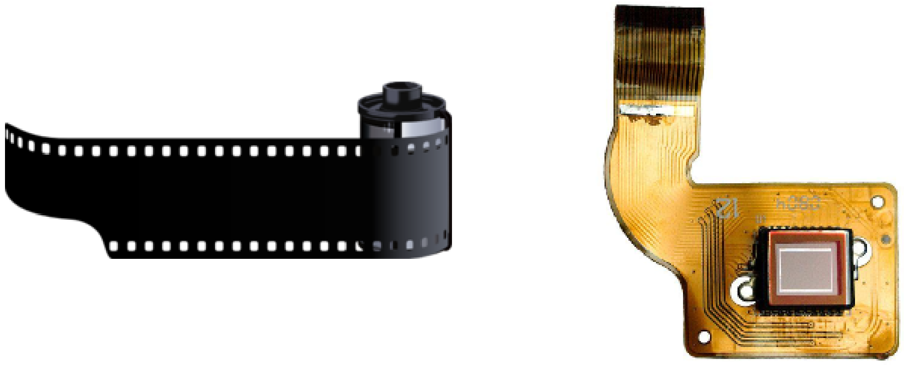


Figure 1.1: In the image above, on the left - chemical photographic film is shown. On the right is a more modern electronic sensor, as present in digital cameras.

leading position of digital cameras in the consumers' hands, several improvements are still possible for these devices.

One important drawback of modern digital censors used in cameras is their limited *dynamic range*. Dynamic range is the ratio between the greatest and smallest values of luminance the camera can understand. Anything beyond the largest value or below the smallest is mapped to the maximum and minimum values, respectively. Unfortunately, the dynamic range of these sensors is many times less than that of the human eyes. What this means is that when viewing certain scenes that contain both bright and dark areas our eyes are able to distinguish all the available details, unlike the cameras' sensors (see Fig. 1.2). Although it is possible to choose which part of the luminance spectrum the camera will "see" by regulating *exposure* (the amount of time during which the sensor is exposed to light) and *sensitivity* (that is, how acute is the sensor's response to a certain quantity of incoming light), it is sometimes impossible to completely capture the luminance variations of a given scene. Because of this, standard photographs taken with digital cameras are sometimes referred to as low-dynamic-range (LDR) or standard-dynamic-range (SDR) photographs.

Many techniques have been developed to deal with this issue. One particularly prominent method is the so called *High Dynamic Range Imaging*, or HDRI, introduced by Debevec and Malik (2). This method consists of taking several pictures of a scene with different exposures, this results in several representations of the same scene, ideally each of which captures a part of the luminance spectrum very well while missing some information in other such segments. These images are then analyzed using previously

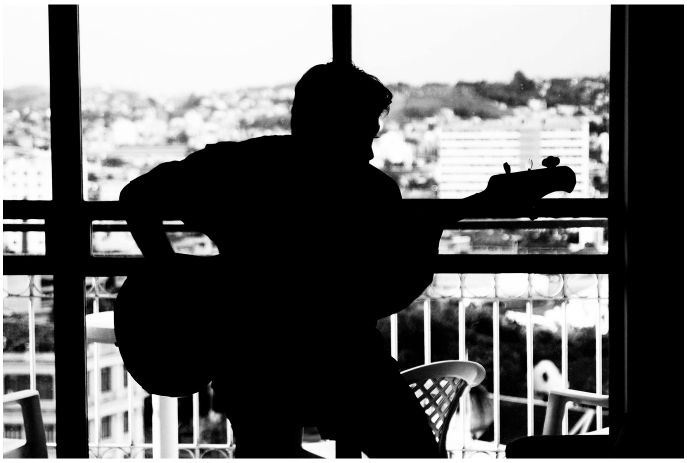


Figure 1.2: A picture showing how the camera's limited dynamic range can sometimes lead to loss of important information on the scene. In this case, the guitar player is completely obscured due to the camera adjusting to a strong background lighting (taken by Luciano Harper).

acquired knowledge of the camera's sensor's response function to obtain a radiance map, which represents the captured scene more accurately than LDR images. The resulting images are called High Dynamic Range images, or simply HDR images. Unfortunately, current viewing devices, such as computer monitors, televisions, projectors and the like are also unable to reproduce the whole gamut of luminance the human eyes can perceive. While HDR viewing devices do exist, they are currently in a development phase and generally not available to the public (3). The solution is thus to artificially reduce the dynamic range of the HDR images back to the usual format supported by regular viewing devices. Such a reduction is done in a clever way so as to make the best possible use of the available dynamic range in viewing devices. This set of techniques is called *Tone Mapping*, for which several differently motivated methods exist. We will discuss HDRI in more detail in Chapter 2. An example of a set of differently exposed images followed by a resulting tone-mapped HDR output can be seen in Fig. 1.3.

Another interesting method for the creation of more pleasant images from a set of differently lit captures is a technique called *Exposure Fusion* (EF) (1). This clever algorithm is a very efficient procedure that produces results similar to those of tone-mapped HDR images through a per-pixel averaging process. While images resulting from the use of EF do not necessarily describe the viewed scene in a realistic fashion

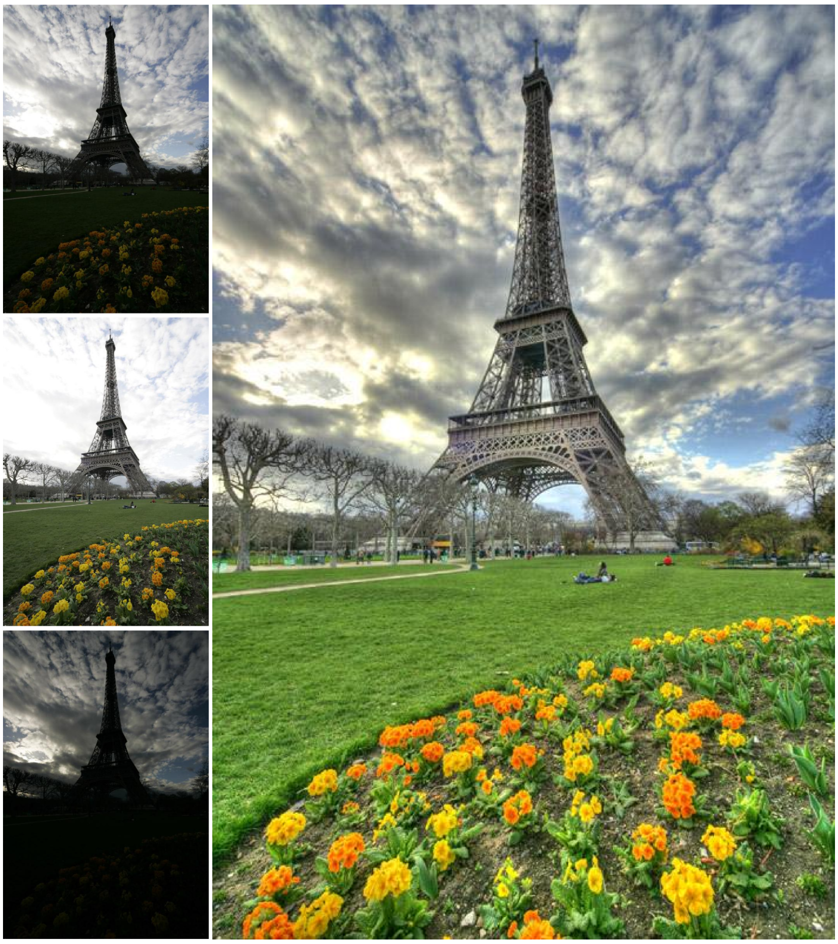


Figure 1.3: An example of the creation of an HDR image from several differently exposed LDR images. The HDR image must then be tone-mapped for a viewable result. Images from Photomatrix.

(the result can have lighting conditions that do not reflect the reality of the scene), this alternative to HDR has many advantages. These include the algorithm's simplicity, its quick processing time (even slow implementations will only take a few seconds to process a set of images), its robustness and the quality of the results. An image depicting the results of EF applied to a set of differently exposed photographs is shown in Fig. 1.4.

In this work, however, we are not interested in the application of HDR or EF to still images. Instead, we wish to study ways of applying these techniques to the creation of videos by users. This adds several new levels of complexity to the problem. There are three main concerns that are particularly troublesome.

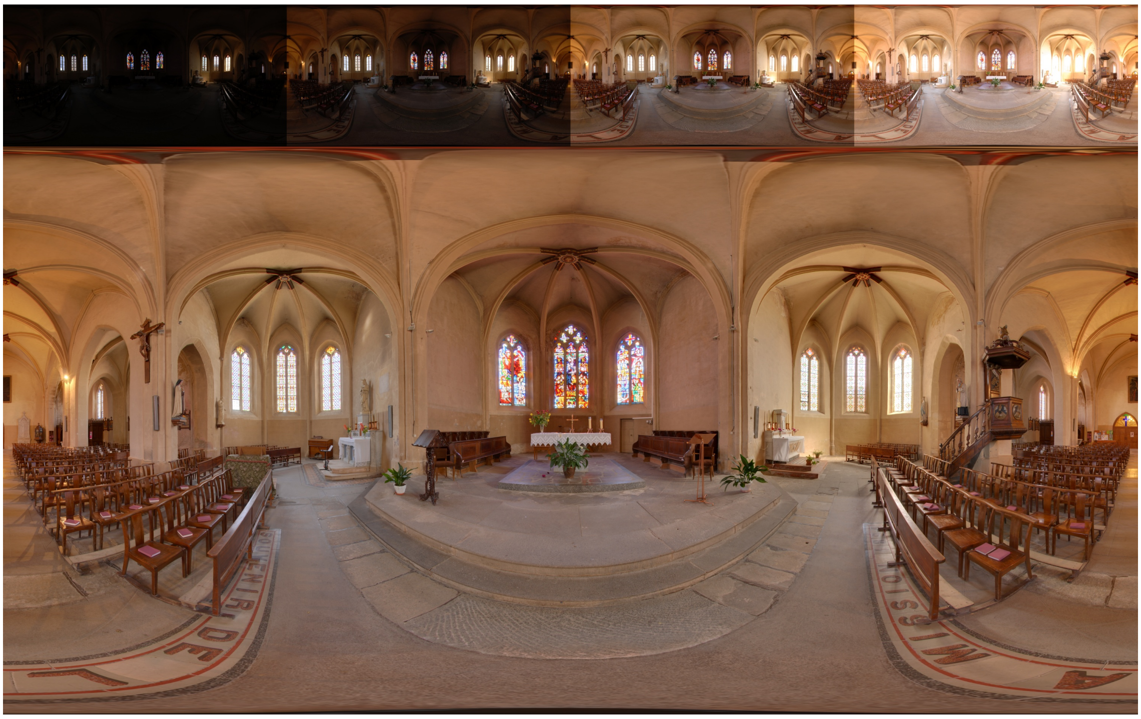


Figure 1.4: This image shows the result of using Exposure Fusion to combine several LDR images into another, albeit more pleasant, LDR image. (Image by Seb Przd).

The first is the acquisition of the images. In fact, in order to obtain the material necessary to create a video with improved dynamic range qualities, it would seem necessary to have a video stream that would involve series of differently exposed images. While it is usually possible to change exposure settings on consumer cameras manually in order to create the input images for HDRI or EF, this would not be possible for video, where the capture rate would be way beyond anything that could be controlled manually. In order to obtain the input videos, a Nokia N900 camera-phone with the Fcam API (4) was used. The acquisition step will be briefly seen in Chapter 3, as it is not the focus of this work.

The other two problems mainly involve movement. Unlike still image processing, video may involve both camera movement on the part of the film-taker and object movement on the scene, for example, people walking. Camera movement proved to be a lesser concern, as methods exist to compensate for this issue. In our case, method (5) was used and provided good results for the vast majority of the cases. This step will be further discussed in chapters 3 and 5, when details of each method are discussed.

Object movement, however, proved to be much more challenging for all techniques involved. For HDR video, the lack of appropriate information about an area of the image is a problem that might generate large unwanted color artifacts. This problem was attacked in our work (6) and then further improved in (7). These and other results are discussed in Chapter 3. For EF, object movement is bound to generate “ghosting” artifacts, that is, the averaging process will even objects out through frames creating what usually seems to be a semi-transparent silhouette. In our work (8) a novel method is introduced to deal with ghosting in EF video. Our results and other considerations will be discussed in Chapter 5.

2

High Dynamic Range—an Overview

2.1 HDR—an overview

This section will explain how HDR works. It will touch the most common reconstruction technique (several exposures), and introduce the camera response curve reconstruction, finding radiance values and tone mapping.

As mentioned in Chapter 1, ordinary consumer cameras are unable to capture the whole dynamic range of scenes visible to the human eye. A new technique was proposed in (2) that allowed the creation of a High Dynamic Range image from the combination of a set of regular Low Dynamic Range photographs. The main focus of (2) is the process of discovering a camera's *response function.*

2.1.1 Recovering the Camera Response Function

Given certain assumptions about the captured images, for instance that they are taken close enough apart so that any changes in the lighting conditions on the scene can be overlooked, we know a digital camera's sensor will map the luminous radiation E_i coming from a certain direction as a pixel value Z_i on the ith pixel in the resulting image (the order of i can be given in any reasonable way). These radiance values, are of course affected by the exposure time Δt_j used to take the photograph. In this way, the value registered by the sensor for each pixel is considered to be given by the product $E_i \Delta t_j$. Thus, the camera response function mentioned above could be defined as f:

$$Z_{ij} = f(E_i \Delta t_j) \tag{2.1}$$

It is natural to assume that f would be a monotone non-decreasing function, after all, a greater amount of energy read by the sensor should not produce a lesser response. However, small variations could be mapped to the same values and too large or small incoming energies are clipped to the minimum and maximum response values, respectively. We will consider, at first, an abstraction of the realistic camera response functions which has infinite precision, thus making the first point irrelevant, i.e., different inputs will be mapped to distinct values. Proceeding to consider only the irradiance values that are not over or under-exposed and knowing that strictly increasing functions are invertible, we can derive the following equations:

$$f^{-1}(Z_{ij}) = E_i \Delta t_j \tag{2.2}$$

Taking the natural logarithm of both sides, we arrive at:

$$ln[f^{-1}(Z_{ij})] = ln[E_i] + ln[\Delta t_j] \tag{2.3}$$

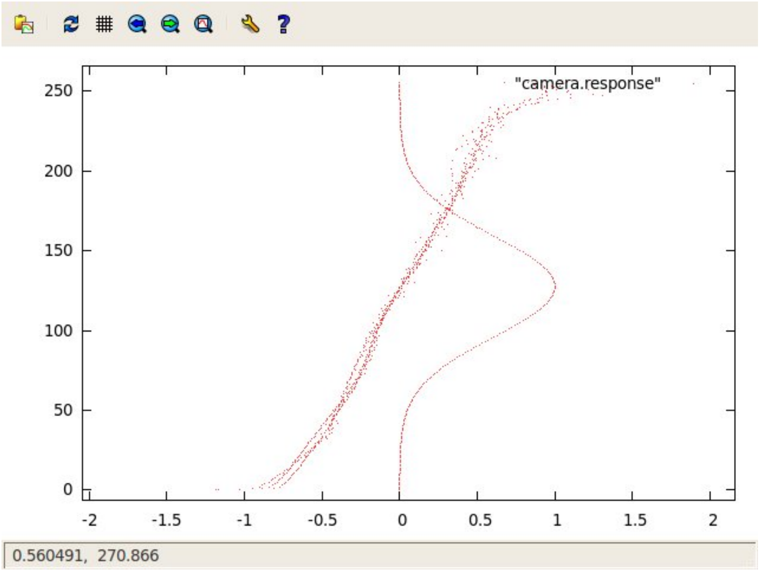
We proceed by defining $g = ln[f^{-1}]$. Thus:

$$g(Z_{ij}) = ln[E_i] + ln[\Delta t_j] \tag{2.4}$$

Given a set of input LDR images, we already know Z_{ij} and Δt_j, and we would like to discover the actual radiance values of E_i, which contain the information present on the image and are outside the boundaries of the camera's otherwise Low Dynamic Range, and the function that maps them into pixel values, g.

Returning to the real version of these functions, with limited precision, considering the domain of g to be pixel values, a finite set, one only has to find a finite number of values of $g(z)$ to fully understand this function. The authors proceed to finding the best possible values of g and E_i to satisfy the previous equations using numerical methods. These steps will not be repeated in this work, but can be found in the original paper (2). The step of recovering a camera's response function is central to HDR. Following it we arrive at a certain discrete function which will allow us to quickly map pixel values to approximate luminance amounts, assuming the exposure times are known for each photograph given to the algorithm and enough information is given as an input to warrant a good approximation of g. It is important to notice that once the response curve of the camera is built, it can be used to recover radiance values for any image taken with this camera using the same sensitivity settings, not just the ones used in the steps described above, as this curve is a parameter of the camera itself (however, it changes in a non-trivial way with sensor gain).

In Fig. 2.1, the response curve for the Nokia N900 camera-phone can be seen. These values were calculated using the *pfstools* package from Max Plank Institut (more information at their website). The second curve appearing on the graph is the weight given to each exposure



0.560491, 270.866

Figure 2.1: This image shows the response curve of a Nokia N900 cellphone. The horizontal axis represents a log-scale of radiosity values, the vertical scale represents pixel values. The bell curve shows the weights given to each pixel value during the process, with lesser weights being given to pixels with poor exposures (the ones closer to minimum and maximum values). This last step is explained below.

2.2 Creating HDR images

In Fig. 2.2, a schematic of a logarithmic scale of luminous dynamic range can be seen. While we now possess the camera's response curve, each LDR photograph taken can only encompass a small part of the scale. Our goal thus becomes to capture as much of the scale present on the scene as possible using several pictures with varying exposure times and then reconstruct the scene together from this information.

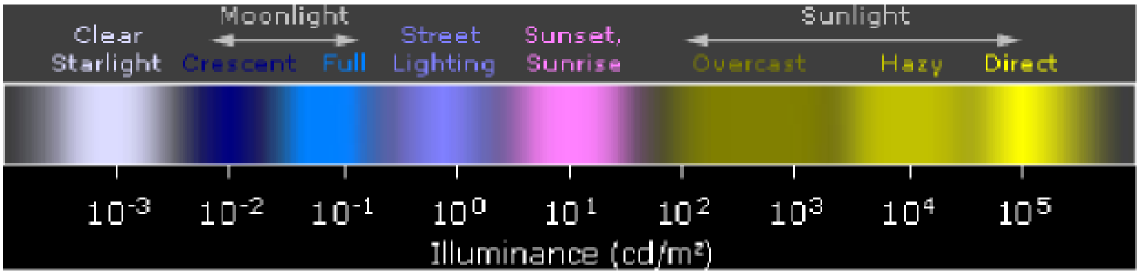


Figure 2.2: This image shows a log-scale axis with Illuminance values, measured in $Candelas/Meter^2$.

In this way, by taking several images of the same scene, we will have a certain pixel correspondence. Considering that the scene's dynamic range exceeds the possibilities of the camera's sensors, this will mean that some pixels will be under or over-exposed, i.e., not quite accurately mapped to the minimum and maximum values possible. Once a pixel is mapped to such a value, we are no longer able to accurately gauge its radiance value, as it might be any number above a certain threshold (which will depend on the camera's sensitivity and exposure time for that shot). In contrast, the same pixel might appear with a non maximal/minimal value in one of the other images. In this way, we can simply take this new value and calculate the intensity of light coming from this pixel's direction with a simple correspondence to the curve shown in Fig. 2.2. In order to obtain the best results, the values taken from each image in the input frames is averaged using a previously selected weighting system, one that will give greater importance to pixels in the more "well behaved" parts of the response curve, that is, the spots closer to the central pixel values. Specifically, from Equation 2.4, we can see that:

$$ln[E_i] = g(Z_{ij}) - ln[\Delta t_j] \tag{2.5}$$

and using all available information on a given pixel from the inputs we arrive at:

$$ln[E_i] = \frac{\sum_{j=1}^{P} w(Z_{ij})(g(Z_{ij}) - ln[\Delta t_j])}{\sum_{j=1}^{P} w(Z_{ij})} \tag{2.6}$$

where $w(Z_{ij})$ is a certain weighting function on the set of possible pixel values. In the original paper by Debevec (2), a simple "hat" function is used. In our work, a bell curve is preferred, as shown in Figure 2.1. For a more thorough analysis of HDR Imaging, we refer the reader to (9).

The result of these operations is that a file is created, where pixels are associated numbers whose values lie in a certain domain of radiance values. These are not in direct correspondence to any set of pixel values, and they are impossible to show on a regular visualization device, such as a computer monitor, cellphone, projector or printer because these displaying devices also have a limited dynamic range. Some options exist to make use of the newly made images: they can be visualized in novel HDR display devices, as seen in (3), or the luminance values can be mapped in a way to fit the regular display device's possibilities. We will focus our attention on the second option. This process is known as *tone-mapping.* The result of a tone mapped HDR image can be seen in Fig. 2.3

Several methods for tone-mapping exist, mainly part of two categories: global and local. Global techniques rely on a single operator applied to all the image's pixels, which are then mapped to a more suitable set. For instance, given a HDR image with radiance values ranging from 0 to N, we wish to map these values to a smaller range 0 to n of available on our display device, where $N >> n$. One simple possibility is mapping the values as:

$$L_{new} = n\frac{L}{L+1} \tag{2.7}$$

Which maps the set $[0, N]$ into $[0, n]$. Unfortunately, while Global tone-mapping methods are usually cheap and easy to implement, they tend to sometimes produce excessively smooth images, resulting in a loss of contrast.

An alternative to global methods are local methods, that is, analyzing a pixel's surroundings in order to achieve a non-uniform mapping which better preserves contrast between regions. Several more complex methods of this kind exist, generating a great variety of results. The particularly interesting method (10) will be briefly mentioned in Chapter 3.

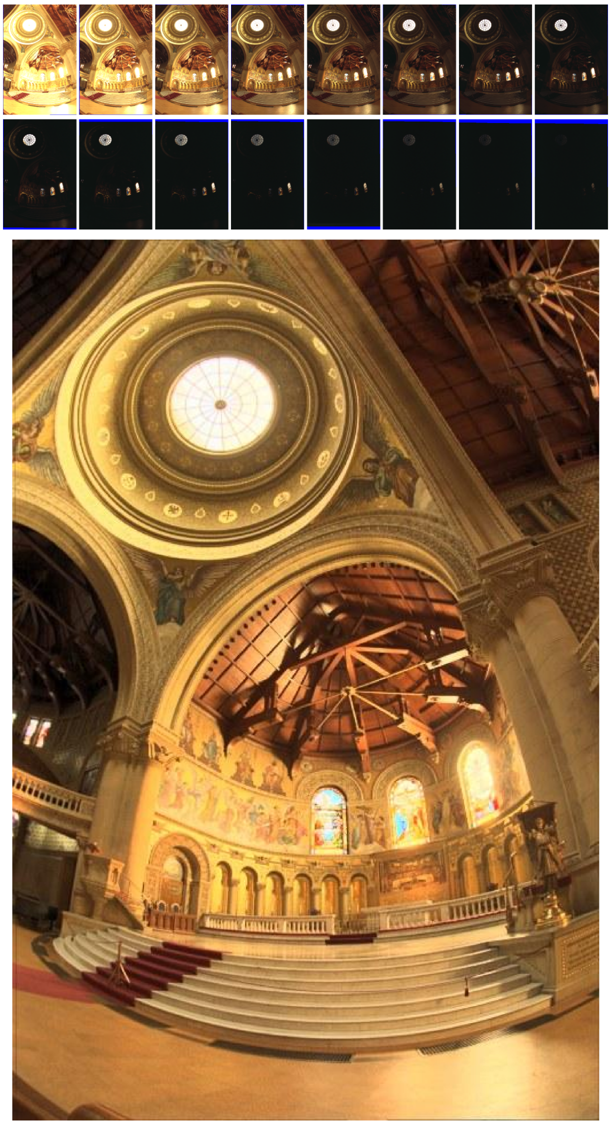


Figure 2.3: A picture showing an example of HDR imaging. Note that none of the input images contain all the details present in the final image, even though it has been tone-mapped to fit the display device's limited dynamic range. Image borrowed from pfstools.

3

Towards Mobile HDR

In this chapter, the results presented in works (6) and (7) will be discussed in further detail.

3.1 Introduction

Since the development of the technique in the late 90's, High Dynamic Range Imaging from pictures with different exposures has never ceased to be a hot topic in Computational Photography. In the beginning, software developed by pioneer researchers (like HDRShop and Photosphere) were released, and soon the feature became available in commercial imaging products such as Photoshop.

Throughout the past decade, several improvements were made in the technique, but the user still had to take several photographs and process them with computer software. Very recently, however, cameras with built-in HDR began to appear. The Pentax K-7, released in 2009, is said to be the first camera featuring HDRI. A more notable example is the iPhone 4 platform, which offers an HDR mode in the camera app since the release of the iOS 4.1 this year.

In the realm of HDR video, cameras with HDR sensors, like the RED Epic and the ARRI Alexa, are being widely used these days in film production. However, such devices are not easily available for the general consumer. An alternative is to use cameras that can capture sequences of frames with different exposures and apply a technique similar to the one used for still images. Fortunately, these cameras are more affordable, but methods for HDR video are still scarce.

In this work we present an HDR video reconstruction method for hand-held cameras, including those installed in some mobile phones. Being based on histograms, the method is more adequate for devices with lower computational power. To the best of our knowledge, previous methods make use of (computationally expensive) optical flow techniques to find correspondences between frames in the photometric calibration phase of the algorithm. These methods assume that the exposure is constant for different frames, therefore and due to this restriction they consider only a subset of all the original captures at one time. In this case the temporal resolution suffers considerably. Our method, in turn, does not reduce the video frame rate.

In the next section, some basic works on HDR Imaging are discussed, especially those that provide more details and background to our approach, which is described in Section 3.3. In Section 3.4 we describe the mobile device implementation. Results are discussed in Section 3.5. Final comments and future directions are shown in Section 3.6.

3.2 Previous Work

The problem of High Dynamic Range image reconstruction for still images is virtually solved (11). Furthermore, many different solutions appeared since the classical work by Debevec and Malik (2). HDR video reconstruction is a natural extension of the image related problem. Therefore, approaches tend to be built upon methods for still images. As it is out of the scope of this text to review all existing approaches, we refer the reader to (9), which is a nice review of the subject.

HDR video reconstruction is a more challenging task because from the hardware side it requires a programmable camera and from the software side the data is dynamic. The earlier reference in this case is (12), where classical vision methods for motion estimation (namely, optical flow) are used to deal with the motion between frames. For a review of methods we refer to (13), where components of the HDR pipeline are presented and discussed with the main focus on video.

Our approach for HDR video is based on histograms. It is efficient, simple and robust to noise. We will discuss the method in Section 3.3. The recovery of the camera response function from images is described with details in (14) and (15). The HDR reconstruction algorithm is a modified version of (16). The histogram-based image registration technique is brought from (5), and the Radiance Map reconstruction with ghost removal is made in a similar way to what is described in (17).

Regarding handheld devices, the problem of HDR reconstruction from misaligned and (possibly) blurred long-exposed photographs is treated in (18). The authors of (4) provide a camera application with HDR mode. In fact they made available a full API for experiments with the low level aspects of the camera hardware. We have used their platform in this work, the video capture being made with a Nokia N900 smartphone.

3.3 HDR Video

The pipeline for HDR Photography from LDR images has two phases: first, photometric calibration must be performed and the radiance map reconstructed to generate the HDR picture, then the dynamic range of the result should be reduced to allow the image to be shown in LDR displays.

In order to recover the camera response function and properly map radiance values, algorithms for HDR rely on pixel correspondences between frames to relate differently exposed values of the same point in the scene. When it comes to video, this is obviously a difficult task.

The solution is to apply some sort of motion estimation. Classical vision methods for motion estimation (such as optical flow), assume that the exposure is constant. However, to obtain HDR video, a sequence where consecutive frames have different exposures should be captured.

To get around this problem, we devised a method where motion estimation is based on histograms. Another advantage of using histograms is that it is less computationally expensive than methods based on optical flow techniques. This is particularly interesting for devices with less capable processors.

Our method has three steps: first, the camera response function is estimated using an histogram-based technique; second, multiresolution alignment of threshold images based on histogram cuts is performed; third, the radiance map is reconstructed observing the variances of radiance values for each pixel. The algorithm is detailed in the following sections.

3.3.1 Photometric Calibration

The input of our algorithm is a sequence of triples[1] of images $\{F^i\}$, where $F^i = \{F_1^i, F_2^i, F_3^i\}$. In our first work (6), exposures were calculated only at the beginning of the capture process, and then kept constant throughout the process. In (7), the method was improved by setting the exposure time for F_1^i as being the optimal value calculated by the camera's auto-exposure feature, while the other two photographs in the set are made with one stop above and below this time, respectively. In this way, F_2^i and F_3^i have exposures that are, respectively, twice and a half of the exposure of F_1^i, for all i.

Algorithms for photometric calibration require a correspondence between pixels of different frames in F^i to be known. We are assuming that exposure changes preserve monotonicity of pixel values. Intuitively, the n brightest pixels in a frame with exposure e_1 correspond approximately to the n brightest pixels in a subsequent frame with exposure e_2, even though their actual values are not the same. Let $\{p_i\}$ and $\{q_i\}$ be the sets of pixels from two consecutive frames (say, P and Q), of the same size, sorted according to the luminance value of the pixel. The radiance mapping $M_{P,Q}$, between P and Q, is defined simply by $M_{P,Q}(p_i) = q_i, \forall i$.

Finally, the actual *pixel value* to *radiance value* mapping can be recovered by applying any of the algorithms available in the literature. For this particular implementation we have used the parametric approach described in (15).

3.3.2 Histogram-Based Registration

Aiming not to reduce the HDR-reconstructed video frame rate when compared to the captured video, we generate an HDR frame for each frame in F^i. Therefore, once i is fixed, for each $j = 1, 2, 3$, the remaining F_k^i should be aligned with F_j^i. We perform a multi-resolution alignment that is described with details in (5).

Roughly speaking, the method is as follows: first, an image pyramid is constructed for each grayscale image exposure. Then, for each level of the pyramid a corresponding median threshold bitmap (MTB) image is constructed. An MTB image has 0's where the input pixel values are less than or equal to the median value and 1's where these values are greater. The overall offset for alignment is computed starting with the lowest

[1] We chose 3, but any reasonably small number larger than 1 could be used.

resolution MTB pair, testing for an offset in the range $\{-1, 0, 1\}$ in the horizontal and vertical directions. At the next resolution level this offset is multiplied by 2 and the result is tested with its pair shifted in the range $\{-1, 0, 1\}$ in the mentioned directions. This continues up to the highest resolution, leading to the final alignment offset.

3.3.3 Radiance Map Reconstruction

The image alignment phase is necessary to deal with camera motion. However, there can also be movement in the scene, and such movement causes ghosting effects during the reconstruction of an HDR frame. We deal with this issue by analyzing the variation of radiance values over the corresponding (aligned) pixels of the images in F^i.

More precisely, for each F^i we reconstruct four HDR images, one for each F^i_j and one considering all the images in the triple. We call them $\hat{F}^i_j$, for $j = 1, 2, 3, 4$, in the mentioned order. Consider we are reconstructing the HDR frame for $\hat{F}^i_1$. Let us pick some pixel p in $\hat{F}^i_1$ and call r_1 its radiance value. Let r_j be the radiance values of the three corresponding pixels across images $\hat{F}^i_j$, $j = 2, 3, 4$. We define σ"2 as the variance of the set $\{r_j\}_{j=1,2,3}$. For the final $\hat{F}^i_1$, the radiance value for the pixel p will be set to a convex combination between r_1 and r_4 depending on the magnitude of σ: if the variance is high, some movement must be happening in this pixel across frames, so we weight the value of the radiance towards r_1; and vice versa if the variance is low. This procedure is repeated for $\hat{F}^i_2$ and $\hat{F}^i_3$.

3.4 Mobile Device Implementation

As discussed in (19), the lack of a fully programmable, portable camera is a problem for Computational Photography researchers. In this sense, the development of the FCam API (4) turns out to be a great step towards a new generation of mobile devices. With the FCam API, it is possible to have full control of the camera parameters, such as shutter speed, gain and focus. We may even change its algorithms, such as autoexposure, demosaicking and autofocus. In this work, we used a Nokia N900 running Maemo 5 (Open Source Linux distribution). We developed a HDR video application which allows the motion of both objects on the scene and the camera.

3.4.1 Capturing HDR Video

The major challenge concerning the capture and processing of HDR frames on a mobile device is to maintain a fine balance between frame rate, memory usage and processing power. On one hand, there is a goal to capture at least 25 frames per second to produce good quality video, and this rapidly consumes the device's memory. On the other hand, the application should process and save all the captured frames, and this stage is a little slower than the first, due to limited processing power and the fact that writing a file on disk requires more time than capturing a photo and saving it in the application memory. Although it is possible to use the method described in Section 3.3 with as many different exposures as desired, the current implementation only uses three different exposure settings. This provides good results without harming this balance. We limit our application to capture only short videos due to these memory limitations. It is a reasonable restriction: mobile devices usually are not designed to capture long duration videos.

The capture stage works as follows: while the user looks through the viewfinder (without recording), the application performs an autoexposure algorithm, corrects the white balance and displays a preview of the scene using the current settings. When the capture starts, the autoexposure algorithm is executed in the background, providing the optimal camera settings for a good exposure during the video capture process. The first shot uses the optimal computed exposure, the second and the third shots use twice and half of this value, respectively. This process is repeated until the video is fully recorded and the frames are sent to the processing stage. All shots use the same gain (which guarantees that they will have the same camera response function, as it varies non-linearly with sensor sensitivity).

3.4.2 Processing HDR video

The processing of the captured frames is done after all the frames were captured. This is a way to guarantee that there is no slowdown during the capture process. Since this stage is done independently of the first, virtually any HDR method could be used here. We chose the method described in Section 3.3 because of its flexibility due to histogram analysis (instead of direct pixel-to-pixel correspondence). This stage could be performed on the mobile device, on another computer, or even on a cloud.

Our current implementation performs this step on a desktop computer for testing purposes; however, it would be possible to perform the full process on the camera itself (instead of only the captures). This stage of implementation is straightforward, since the same code can run on both machines.

3.5 Results and Discussion

As was mentioned in Section 3.4, in the current implementation stage the algorithm being used on the Nokia N900 returns a sequence of images with varying exposure times. We then proceed by transferring the results to a desktop computer in order to process the data. This is done by applying the algorithm described in Section 3.3 to the images. After this step, we have a set of HDR images, which correspond to each frame of the captured video. In order to be able to visualize our results on regular LDR devices, a tone-mapping algorithm is also necessary. This step is done by using the `pfstmo` library, from Max Plank Institut[1]. After testing several tone-mapping methods, we have decided to use an implementation of the method (10) present in the library, due to both its speed and the quality of the results obtained. The final resulting tone-mapped output, along with a sequence of three differently exposed frames generated by our program can be seen on Figures 3.1 and 3.2.

Figure 3.1 shows the results obtained in an outdoor scene, with predominant camera movement. Notice that the background (resp., the building interior) is only well-exposed on the second (resp., third) captured frame. Both areas of the image are well shown in the three tone-mapped results. Figure 3.3 shows another result.

Figure 3.2 shows the results obtained capturing an indoor scene, with object and camera movement. Notice the movement between the captured frames.

3.6 Conclusions and Future Work

In its current form, the algorithm works as follows: firstly, the Nokia N900 with an FCam API is used to capture a sequence of frames with varying exposures. These frames are used on a desktop computer to generate a series of HDR images, which are then converted to regular LDR images through a tone-mapping algorithm.

[1] Available at http://www.mpi-inf.mpg.de/resources/tmo/

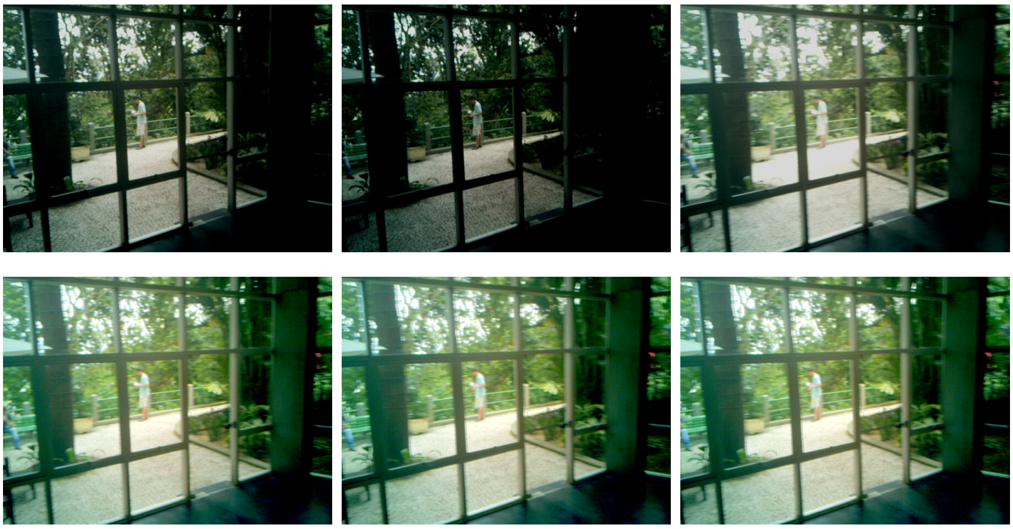


Figure 3.1: (Top) Captured frames. From left to right: optimally-, sub- and super-exposed shots. (Bottom) Corresponding results with tone-mapping.

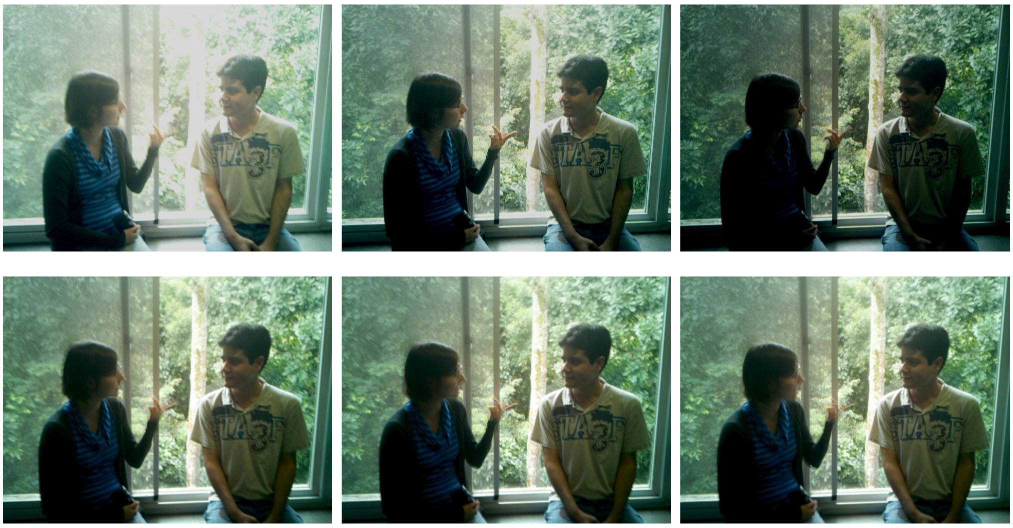


Figure 3.2: (Top) Captured frames. From left to right: sub-, normal- and super-exposed shots, according to initial calibration on a brighter scene. (Bottom) Corresponding results with tone-mapping.

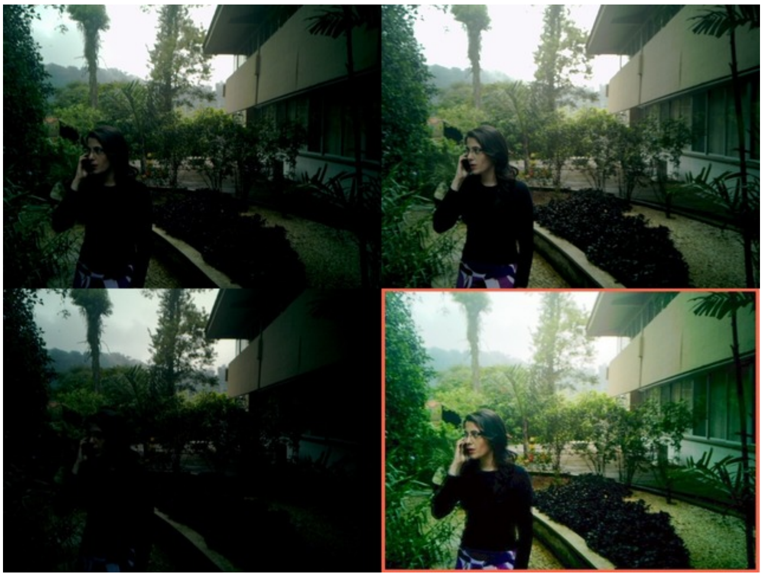


Figure 3.3: Another set of results. The HDR image is presented in a red frame. Notice the details in the sky, clothes and vegetation, missing from the original images.

An improvement might come from other means of pixel correspondence for object movement on the scene, such as optical flow, to further enhance the quality of the results. Also, other Tone Enhancement techniques could be used to improve the quality of the captured videos.

A more difficult challenge for future works involves finding a better way to deal with the device's small memory, in order to increase the amount of frames that can be captured. A possible partial solution lies in creating a low priority thread that would save images to the device's hard disk while the program is still running.

Finally, we believe that the use of a fully programmable camera brings many more possibilities besides the ones that have been explored here. Many ideas involving user-aided capture and processing, or the retrieval of geometric proprieties of objects through intelligent capture processes, can be achieved by having access to low level hardware parameters, as was done in this work.

4

Alternative Methods - Exposure Fusion

In this chapter we will discuss *Exposure Fusion*, a technique first presented in (1). Unlike the introductory chapter, more technical details will be presented here, followed by our own work involving improved video using this algorithm in Chapter 5.

Unlike HDR, EF is not physically-based. The authors are not concerned with capturing the real details of the presented scene, but rather want to simply obtain an image that possesses the desirable qualities of a tone-mapped HDR image. This process is achieved without ever actually calculating any sort of camera parameters or estimating the camera response, but uses only a clever per-pixel averaging process. The main advantages of Exposure Fusion over HDRI involve its speed and simplicity. The algorithm is also very robust under difficult circumstances, which is particularly interesting from the point of view of improving videos - as a low capture framerate means that each inbound image has very low redundancy in regards to the other frames. A comparison of HDRI and EF for a still scene can be seen in Fig. 4.1.

4.1 Exposure Fusion Pixel Weighting System

Exposure Fusion's method can be understood by imagining the incoming images with different exposures as a stack, which is vertically averaged on a per-pixel basis, using a weight map for each image based on certain criteria. These criteria are as follows:

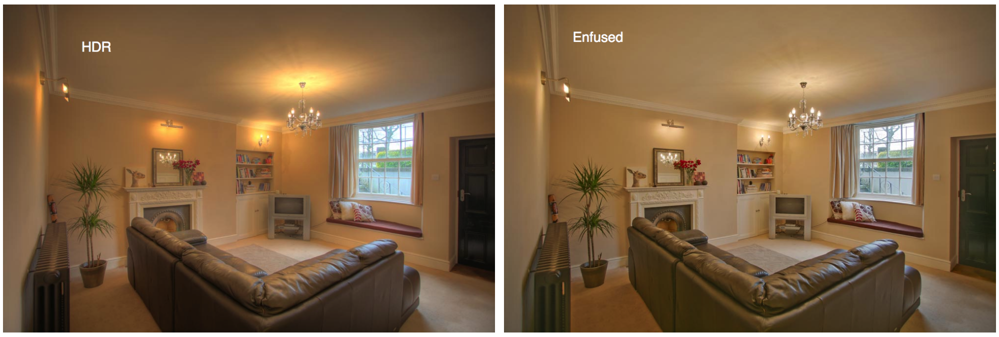


Figure 4.1: A comparisson between tone-mapped HDRI (left) and Exposure Fusion (right) created using commercial software, taken from Digital Photography School.

1. *Contrast:* A high-pass Laplacian filter is applied to the grayscale version of the images. This gives greater values to high-frequency pixels, such as those on borders and textures. This weight, C, helps preserve the sharpness of the images. It is interesting to notice that higher exposed images might be inherently more blurred than their low-exposed counterparts from some amount of camera movement occurring during the exposure, and the Contrast measure also helps reduce theis effect.

2. *Saturation:* Longer exposure times usually result in desaturated colors, which are less pleasant to the viewer and can make the image lose vividness. A saturation measure S is introduced as the standard deviation of each pixel on the R, G and B channels.

3. *Well-exposedness:* In order to avoid over and under exposed pixels, a well-exposedness parameter E is introduced. It is based on the proximity of the pixel's R, G and B values to 0.5 on a 0 to 1 scale when graded by an exponential curve: $exp(-\frac{(i-0.5)^2}{2\sigma^2})$, where σ is an empiric constant, normally 0.2. The results for each color are then multiplied to yield the final weight map.

An example can be seen below in Fig. 4.2. In addition to the weight maps, the authors propose exponent parameters to better control each weight map, i.e.,

$$W_{ij,k} = (C_{ij,k})^{\omega_C} \times (S_{ij,k})^{\omega_S} \times (E_{ij,k})^{\omega_E} \tag{4.1}$$

The parameters ω_C, ω_S and ω_E are usually set to 1, effectively going unused. In some applications, however, giving a greater or lesser power to some of the weights may prove useful.

Once the weight values are computed, the images are fused. This is an averaging process, where a linear combination of the corresponding pixels on each input image is created using the weight maps. These maps are normalized beforehand, so that the linear combination is convex, that is:

$$W'_{ij,k} = \frac{W_{ij,k}}{\sum_{k'=1}^{N} W_{ij,k'}} \tag{4.2}$$

With the fusion correspondingly being:

$$R_{ij} = \sum_{k=1}^{N} W'_{ij,k} I_{ij,k} \tag{4.3}$$

Where I_k is the k-th image in the input sequence.

4.2 Additional Considerations

Unfortunately, simply doing a convex linear combination of the images does not render good results. Quick variations on the weight maps result in undesirable seams on the output. To solve this problem, the technique explained above is applied to a Laplacian Pyramid decomposition of the input images. This means the fusion occurs on each separate resolution level and afterwards these levels are blended together. The result of this operation is a smooth and pleasant image, as seen in Fig. 4.2. Further discussion on this matter can be seen in the original work (1).

Last but not least, EF can also be efficiently used to fuse images with very different lighting, capturing the best parts of each lighting situation. While HDRI requires that illumination on the scene remains rather constant, changing only linearly due to the varying exposure times, EF can be used, for example, to combine images with and without flash. An example can be seen in Fig. 4.3.

An important detail is that the input images must be well aligned. This step is usually performed by a registration algorithm applied beforehand. Should the inbound images be misaligned due to, for instance, camera movement during the capture, pixels combined during the fusion process might correspond to different points on the scene, generating an undesirable effect. In this work, the same registration algorithm was used as in Chapter 3, (5). This simple algorithm once again allows us to eliminate errors due to camera movement—something quite common in mobile video captures. A greater challenge, however, is object movement on the scene during the capture process of the input images. Moving objects from one frame to the next are averaged by the fusion process, which results in *ghosting* artifacts. An example of heavy ghosting can be seen in Fig. 4.4. This issue is especially crucial for video processing, as object movement on the scene is greatly present in most videos.

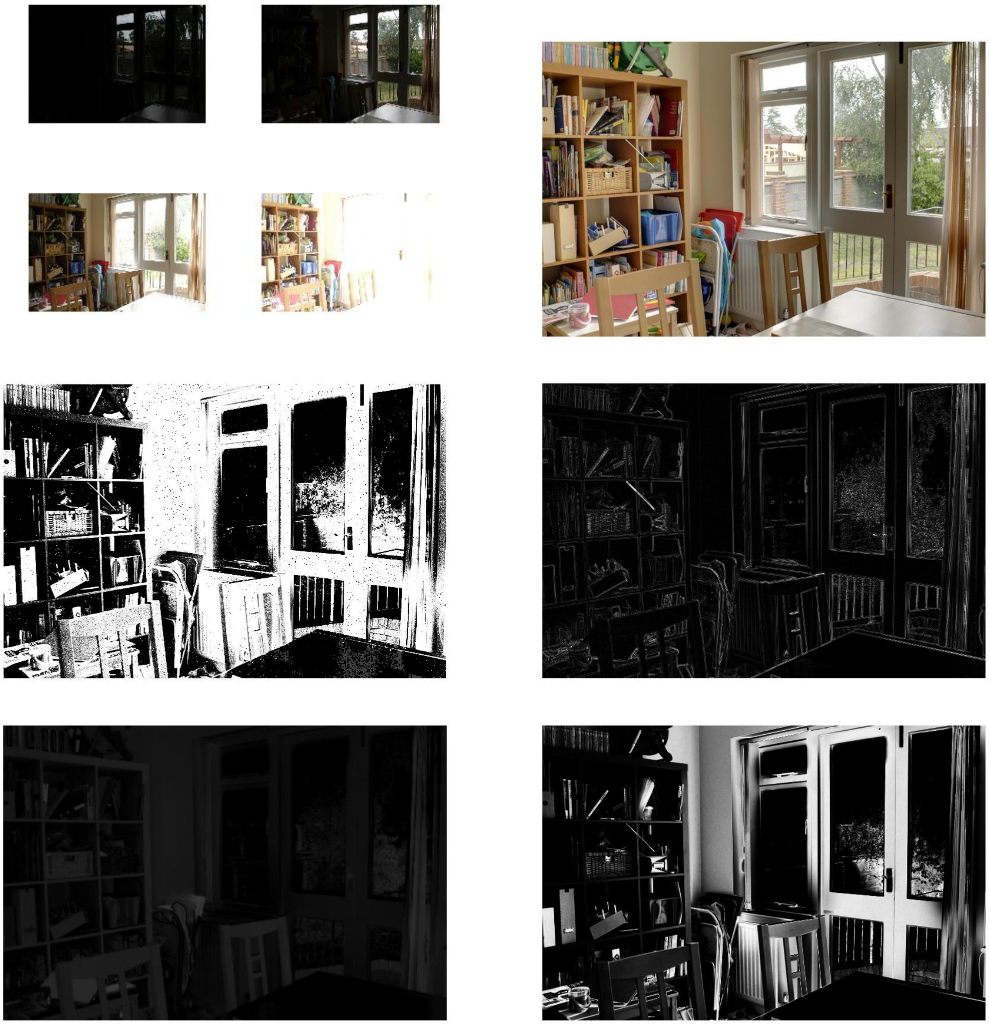


Figure 4.2: (Top Left) Four images with varying exposures used as input.
(Top Right) Resulting Exposure-Fused image.
(Center Left) Final weight map for the third image in the stack, obtained from the product of the three weight maps below.
(Center Right) Weight map for the Contrast parameter for the third image in the stack.
(Bottom Left) Same for the Saturation parameter.
(Bottom Right) Same for the Well-exposedness parameter.

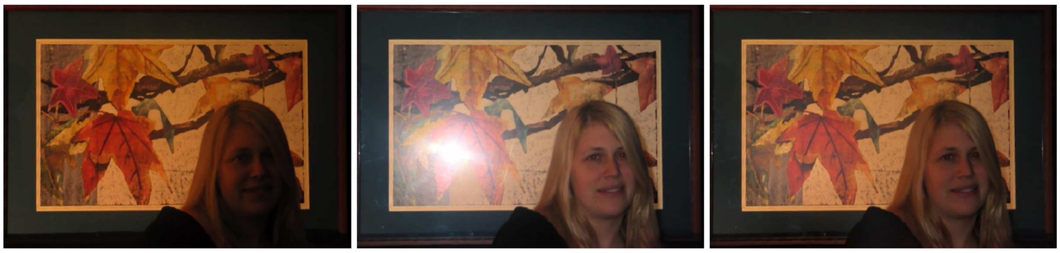


Figure 4.3: An example of a regular photograph (left) fused with a photograph taken with flash (right). Notice how the over-exposed glare on the painting's glass is removed by the algorithm using the Well-exposedness parameter. Image borrowed from (1).

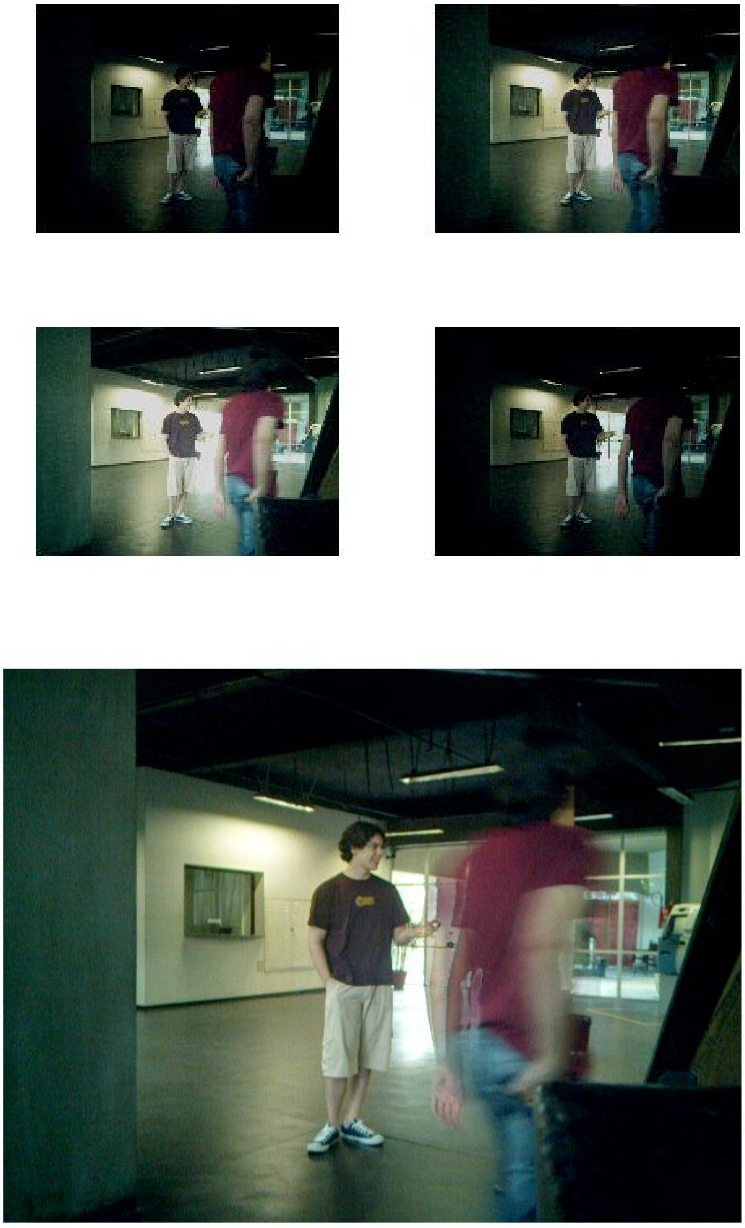


Figure 4.4: An example of ghosting caused by object movement on the scene. The moving man in the input images (left) is averaged through the fusion process and has a transparent appearance in the final image (right).

5

Exposure Fusion Video

While HDR videos have been somewhat explored, Exposure Fusion has seldom been applied to video processing. It is, however, a promising prospect for the creation of improved quality material.

In this section, we will present the results published in (8). As was explained in the last section, ghosting is a major issue with exposure fusion, when used to create scenes involving moving objects. This is particularly important when considering the creation of improved mobile videos, as these are usually expected to contain all kinds of movement. We present a novel method that deals with the elimination of such artifacts by using several carefully selected filters and performing a local analysis. In order to tackle the variations brought by the changing exposure times between frames, features that are known to stay reasonably constant in this case—edges, saliences and textures are used, as detected by a high-pass filter. This method was inspired by our past work dealing with HDR video on mobile devices (7) and a deghosting method for HDR that involved the use of pixel regions as an estimator (20).

5.1 First Attempts

This section will discuss a first attempt at creating a novel deghosting method to deal with some of the issues that surfaced in the creation of Exposure Fusion based video. The method is based on pixel regions and is outlined below.

As seen in (1), applying the EF algorithm to a set of images involves a sum of pixels weighted with certain coefficients. The proposed method involves the addition of yet another coefficient that is used separately from the other three. This value is called the deghosting coefficient and is intended to remove areas that encompass moving objects from the final result, with the exception of frames where the objects are in the same position as those in the frame being taken as reference, thus removing ghosts.

As in the original method, the deghosting coefficient is a floating point number assigned to each pixel in the stack of images used to create an improved video frame. This coefficient is obtained using only the grayscale values of the original images' pixels (in the future using the Luma part of the images in the YUV colour space may be more desirable, as the currently used Nokia N900 camera-phone saves its images in this space automatically).

First of all, a gaussian low-pass filter is applied to all of the used images in order to filter out undesirable noise. Following, in order to obtain the coefficient of a certain pixel (i, j), a 3×3 region centred on this pixel is taken on both images. The size of these regions was obtained through trial and error and seems to work better than larger areas, that appear to capture too much information and are unable to reliably discover movement occurrences. This size could however be larger, depending on the application. In the future, machine learning can be utilized in order to figure out what kind of mask will work best with a certain scene.

The obtained 3×3 matrices are then divided by the sum of all their elements, in order to obtain more even masks with overall element sum being 1. This makes the patches taken from the images more even, as they usually come from different exposures.

Next, one image whose movement we want to preserve must be chosen. For instance, given images 1 through N as an input, if we wish to obtain the Exposure Fusion video frame related to image number 1, we will assign a deghosting weight to pixel (i, j) for images 2 through N based on the difference between the first and the subsequent patches taken. The coefficient is taken as:

$$K(i, j) = 1 - |(\sum_{i=2}^{N} P_1 P_i)^{0.1}|$$

The exponent coefficient was obtained from manual testing and gives a good balance between object movement detection and small differences due to the normal distinction of the differently exposed images.

The resulting coefficient can be seen below as an example. Here the numerical values are shown as grayscale.

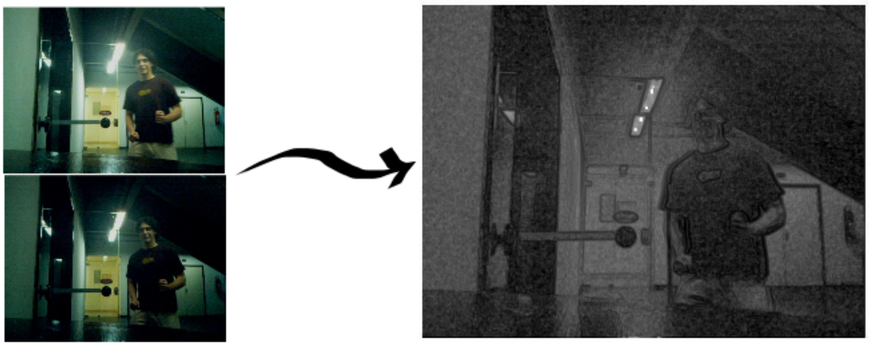


Figure 5.1: Simple calculation of absolute difference between pixel regions.

Unfortunately, the deghosting parameter also detects the normal variations due to the difference of exposure between the frames. This detection is especially strong in regions that are poorly exposed. The result of this detection is that image backgrounds are taken with very different weights in the Exposure Fusion process, which causes the resulting video to have flickering background lighting.

While not completely solving the problem, several measures were taken to dramatically reduce this undesirable effect. Particularly, using the previously calculated exposure parameter, a cut-off has been placed that sets to zero the deghosting parameter values for pixels that are already poorly exposed. The reasoning behind this is that poorly exposed regions will already not be considered by the algorithm from (1), and thus need not receive any further consideration.

Another cut-off is placed on pixels that have small deghosting coefficient values. This is done because pixels where actual movement took place usually have larger deghosting coefficient values, which in turn means that the smaller values belong to background pixels and are due to the expected differences between the images.

An image of the resulting coefficient values making use of the thresholds introduced above can be seen below in Fig. 5.2. This is an altered version of the image above. A few constant factors are used in the created method in order to set the cut-offs and are currently selected manually, based on testing.

Finally, the image sequence is rebuilt, making use of the deghosting coefficients created in the earlier steps during the making of a frame, to filter out the moving

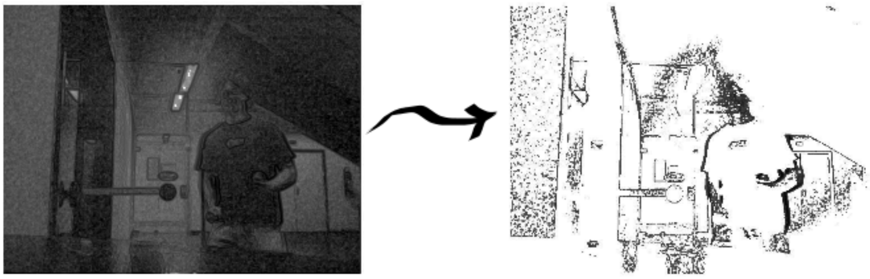


Figure 5.2: Previously seen coefficients after processing.

objects from all frames except the one currently being created. The resulting images have reduced ghosting effects, which sometimes leads to great improvements in the images' visual pleasantness.

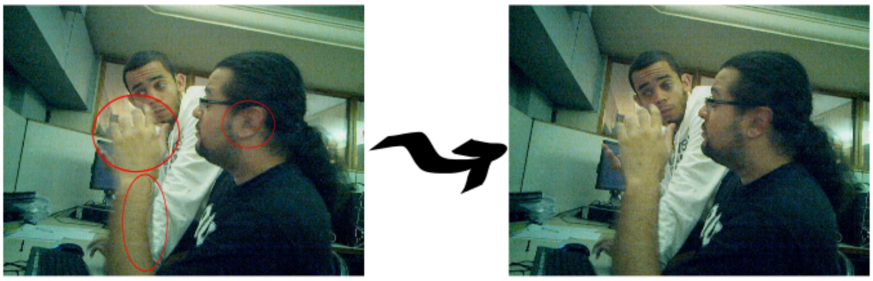


Figure 5.3: A favorable result of the algorithm.

Unfortunately, this approach proved to be less than optimal due to the heavy influence of the non-movement pixels remaining after the threshold filtering.

5.2 Deghosting Method

Following the initial attempts at Exposure Fusion Video deghosting, a second path was taken to ameliorate the resulting videos.

As before, the original images' pixel color variations were found too steep due to the implicit exposure variation to generate reliable results. Because of this, the process outlined below is applied to the result of a regular High-Pass Laplacian filter applied to each image, described as a function $HP(I_k)$. The use of a High-Pass filter eliminates regions not containing high-frequencies, which means that luminance variations due to varying exposures are greatly reduced (see Fig. 5.4).

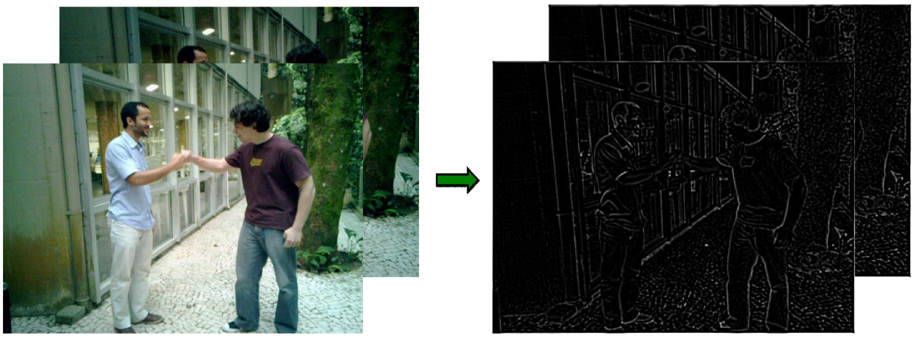


Figure 5.4: Input images shown on the left side, the (amplified x5) result of a High-Pass Laplacian Filter on the right.

Thus, to find the Ghosting parameter of pixel (i, j), $G(i, j)$, we analyze the regions A and B as given by $(i - l : i + l, i - l : i + l)$ in each image.

The process is then repeated for each pixel using a different input. The initial images I_1, I_2 are now subject to a Low-Pass Gaussian filter, given by $LP(I_k)$. Following, we take $HP(LP(I_k))$ as our inputs and proceed to analyze their pixel regions. This method is then repeated with additional Low-Pass filter steps. The pixel areas are evaluated according to the following formula: $G(i, j)_i = 1 - \sum_{n=1}^{2l} \sum_{n=1}^{2l} |A - B|_{n,m}$, that is, the simple absolute diference between these regions.

The resulting obtained Ghosting coefficients are multiplied to obtain the final pixel Ghosting parameter value. This process attenuates the contribution of capture noise and irrelevant weaker high-frequencies, which disappear after consecutive Low-Pass applications, resulting in less erroneous detections of non-movement high-frequency variations and a strengthened Ghosting parameter for pixels that involve true object movement (See Fig. 5.5).

Finally, the ghosting parameter is used in the same way as the three initially proposed coefficients and allows us to reduce the contribution of pixels from I_2 that show moving objects in relation to I_1.

This process is then repeated for consecutive pairs of images I_k, I_{k+1}, $k = 1, ..., n - 1$ to form the final video. The whole process is outlined in Fig. 5.6. More information is available at the author's website, *http://www.impa.br/ achapiro/deghost*, where a video is also presented.

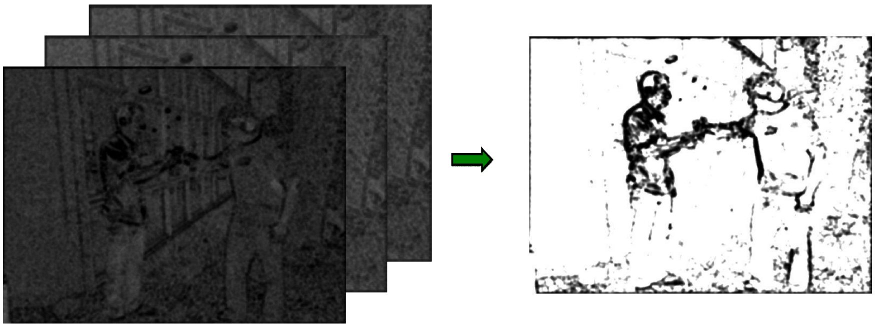


Figure 5.5: On the left - resuls of the local analysis for several Low-Pass applications. On the right, the final deghosting parameter.

5.2.1 Acquisition and Registration of Input Images

In order to obtain the video frames used in this work, a Nokia N900 running Maemo 5 and the FCam API was used, with the same configuration as the one presented in Chapter 3. We proceed to perform a multi-resolution alignment based on image pyramids. This step is necessary, as background pixel correspondence is crucial for this work. Both are explained in detail in our previous work (7).

5.3 Results and Future Work

Testing of the algorithm showed good results when applied to regular situations, removing most of the ghosting artifacts and resulting in improved video quality. Some issues arise with quick movement relative to the camera's capture rate, where sometimes ghosting is not properly treated.

Future work may include improvements to the algorithm's robustness to quick object and camera movement as well as improvements of the filter-based deghosting technique through the use of multi-resolution to aid in the location of shifting objects. One idea consists of comparing the patches in the vicinity of the target pixel for similarities with the original pixel's neighborhood. This means that given images as an input, when we build the pixel (i, j) on the fused image, instead of analyzing the regions surrounding the pixels (i, j) on the initial images, we would also analyze the neighboring pixel's surroundings. Should a region with greater similarities be found, i.e. a smaller ghosting coefficient, we could consider that this pixel would be a better bet for the fusion algorithm and perform EF corresponding it to the (i, j) pixel in the original

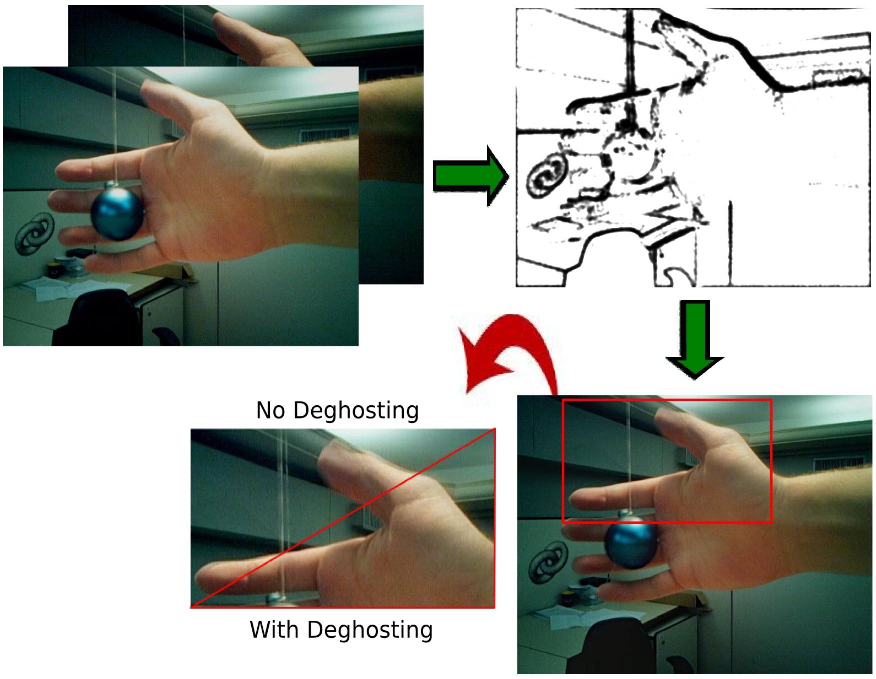


Figure 5.6: Video Creation Pipeline: (top-left) Inputs, (top-right) Ghosting Coefficients after High-Pass filtering (bottom-right) Deghosted result (bottom-left) Comparison.

image. This procedure would then approximate a rough optical flow technique, when done in multiresolution.

Another interesting line for future works would involve the application of the EF algorithm several times, with the new input being the output of the last application. This would further reduce the strobing effect in the background and perhaps introduce details from different light settings into the current frame, while also not overloading the deghosting algorithm with too many input frames that are very distinct from each other.

6

Conclusions

As a conclusion to this work, we must state that both High Dynamic Range Imaging and Exposure Fusion are powerful techniques and their use to video processing can bring very pleasant results. These areas are, however not quite well explored at the moment.

In Chapter 3, our works (6) and (7) are explained in greater detail. These results show that it is possible to create HDR video using a handheld device. More importantly, they present the possibility of creating HDR video on a per frame basis, that is—creating HDR video with an output of the same size as the input, relative to the amount of frames. While some problems still persist, namely that the low amount of information creates either blurry object movement or the possibility of numerical errors during radiance value recovery, HDR video can be a viable addition to future cameras, and specifically cell-phones, such as the one on which this work was implemented.

Our novel deghosting method for Exposure Fusion video presented in (8) and explained in Chapter 5 is another interesting option for mobile devices. Since Exposure Fusion is a much cheaper method than HDR, it might be particularly well suited for camera-phones. In fact, as demonstrated in (21), Exposure Fusion can already be approximated on the Nokia N900 cellphone using an FCam API in real time, which would suggest that Exposure Fusion video could also be built on the same platform with reasonable timing for daily usage. The low framerate of captured videos (25 frames per second) on the Nokia N900 caused additional problems for our method, as movement between adjacent frames becomes greater. Using a higher framerate (such as that of a consumer video camera) would make faster movements less likely to cause ghosting even after applying our method. An immediate solution might involve the use of the larger Frakencamera prototype presented in (4), which was unfortunately unavailable during the crafting of this work.

A more generalistic idea for future works involves the attempt of using the capabilities of the camera-phone to the fullest extent, trying to retrieve and combine as much of the information related to the scene as possible. Specifically, a combination of the pixel's radiance values retrieved using HDR techniques, diverse pixel parameters obtained from EF and perhaps other luminous or even geometrical informations can be used to attempt to re-create the scene in a more complete sense, which in turn could be later used for more advanced visualizations.

References

[1] T. Mertens, J. Kautz, and F. Van Reeth. **Exposure Fusion: A Simple and Practical Alternative to High Dynamic Range Photography**. *Computer Graphics Forum*, **28**(1):161–171, 2009. vi, 3, 23, 25, 28, 31, 33

[2] Paul E. Debevec and Jitendra Malik. **Recovering high dynamic range radiance maps from photographs**. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 369–378, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. 2, 7, 8, 11, 14

[3] Helge Seetzen, Wolfgang Heidrich, Wolfgang Stuerzlinger, Greg Ward, Lorne Whitehead, Matt Trentacoste, Abhijeet Ghosh, and Andrejs Vorozcovs. **High Dynamic Range Display Systems**. In *Proc. of SIGGRAPH '04 (Special issue of ACM Transactions on Graphics)*, August 2004. 3, 11

[4] Andrew Adams, Eino-Ville Talvala, Sung Hee Park, David E. Jacobs, Boris Ajdin, Natasha Gelfand, Jennifer Dolson, Daniel Vaquero, Jongmin Baek, Marius Tico, Hendrik P. A. Lensch, Wojciech Matusik, Kari Pulli, Mark Horowitz, and Marc Levoy. **The Frankencamera: an experimental platform for computational photography**. *ACM Trans. Graph.*, **29**(4):1–12, 2010. 5, 15, 17, 40

[5] Greg Ward. **Fast, Robust Image Registration for Compositing High Dynamic Range Photographs from Handheld Exposures**. *JOURNAL OF GRAPHICS TOOLS*, **8**:17–30, 2003. 5, 15, 16, 26

[6] Tassio Knop, Alexandre Chapiro, Marcelo Cicconet, and Luiz Velho. **Towards Mobile HDR Video**. In *International Conference on Computational Photography: ICCP 2011 posters*, 2011. 6, 13, 16, 39

[7] Tassio Knop, Alexandre Chapiro, Marcelo Cicconet, and Luiz Velho. **Towards Mobile HDR Video**. In *Eurographics 2011*, 2011. 6, 13, 16, 31, 36, 39

[8] Alexandre Chapiro, Marcelo Cicconet, and Luiz Velho. **Filter Based Deghosting for Exposure Fusion Video**. In *SIGGRAPH '11: ACM SIGGRAPH 2011 posters*, 2011. 6, 31, 39

[9] Asla M. Sa. *High Dynamic Range Image Reconstruction.* Morgan & Claypool Publishers, 2008. 11, 14

[10] F. Drago, K. Myszkowski, T. Annen, and N. Chiba. **Adaptive Logarithmic Mapping For Displaying High Contrast Scenes**. *Computer Graphics Forum*, **22**:419–426, 2003. 11, 19

[11] Luiz Velho. **Histogram-based HDR video**. In *SIGGRAPH '07: ACM SIGGRAPH 2007 posters*, page 62, New York, NY, USA, 2007. ACM. 14

[12] Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. **High dynamic range video**. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 319–325, New York, NY, USA, 2003. ACM. 14

[13] Karol Myszkowski. *High Dynamic Range Video.* Morgan and Claypool Publishers, 2008. 14

[14] M.D. Grossberg and S.K. Nayar. **Determining the Camera Response from Images: What is Knowable?** *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**(11):1455–1467, Nov 2003. 14

[15] T. Mitsunaga and S.K. Nayar. **Radiometric Self Calibration**. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **1**, pages 374–380, Jun 1999. 14, 16

[16] Mark A. Robertson, Sean Borman, and Robert L. Stevenson. **Dynamic Range Improvement Through Multiple Exposures**. In *Proceedings of the IEEE International Conference on Image Processing*, **3**, pages 159–163, Kobe, Japan, October 1999. IEEE. 14

[17] Tae-Hong Min, Rae-Hong Park, and SoonKeun Chang. **Histogram based ghost removal in high dynamic range images**. In *ICME'09: Proceedings of the 2009*

IEEE international conference on Multimedia and Expo, pages 530–533, Piscataway, NJ, USA, 2009. IEEE Press. 15

[18] Pei-Ying Lu, Tz-Huan Huang, Meng-Sung Wu, Yi-Ting Cheng, and Yung-Yu Chuang. **High dynamic range image reconstruction from hand-held cameras**. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, **0**:509–516, 2009. 15

[19] Marc Levoy. **Experimental Platforms for Computational Photography**. *IEEE Computer Graphics and Applications*, **30**:81–87, 2010. 17

[20] Gallo, Orazio and Chen, Wei and Gelfand, Natasha and Tico, Marius and Pulli, Kari. **Artifact-free High Dynamic Range Imaging**. *ICIP 2009.*

[21] Natasha Gelfand, Andrew Adams, Sung Hee Park, Kari Pulli. **Multi-exposure Imaging on Mobile Devices**. *ACM Multimedia 2010.*