1 Introduction

When examining the world around us, it becomes apparent that the lighting conditions in many scenes cover a great dynamic range which we are unable to capture with ordinary photography. From shadows to fully lit regions and possibly direct views of light sources, the radiance of objects in a scene could span many orders of magnitude.

It is not unusual that the dynamic range in scene is 100,000:1 or more. Using an ordinary digital camera with 256 quantization levels per color channel it is obvious that we cannot cover such a dynamic range. To compensate for this we use High Dynamic Range Radiance Maps which are generated from a set of digital photographs of a scene captured with varying exposure time. Choosing the exposure times and aperture such that we cover the entire dynamic range of the scene we can construct an HDR image containing a true measure of the relative radiance\(^1\) in the scene.

This practical is about capturing and generating High Dynamic Range Radiance Maps, HDR images. The goal is to implement and understand the HDR algorithm.

The tutorial for this lab is written for Matlab, but you can of course choose to implement the algorithm using any programming language of choice. To your help you have a number of documents and papers listed as suggested reading on the Image Based Rendering practicals home-page and lecture notes. In this document you will also find a brief overview of the HDR algorithm and some notes on digital cameras. A lot of information on High Dynamic Range Images can also be found at http://www.debevec.org. There you can also find HDRshop, a high dynamic range image software for assembling and editing high dynamic range images.

\(^1\)Radiance can be thought of as the number of photons arriving per time at an infinitesimal area from a certain direction. Radiance describes the intensity of light at a point in space from a given direction. Radiance represents the light of an object when detected by a light sensitive sensor device such as a camera or the human eye.
For this lab you will need:

- A set of low dynamic range images
- The camera curve from the camera used to capture the low dynamic range images
- Matlab scripts to make the lab easier

The data and scripts needed can be downloaded from the practicals home-page.

2 Digital Cameras and High Dynamic Range Images

When a photo of a scene is taken with a digital camera or scanned from a photograph the result is an array of brightness values, a digital image. These brightness values are usually not a true measurement of the relative radiance in the scene, i.e. the actual observed irradiance value for a pixel that has twice the value of another is unlikely twice the value of the irradiance observed for the other one.

In image based rendering and lighting methods the images used are often assumed to be accurate measurements of the relative radiance in the scene. This is important since the goal is to as closely as possible describe the real world.

2.1 Digital Cameras

In a digital camera the image is captured with a charge coupled device (CCD). It has an array of pixels that gets charged according to the amount of incident radiance at a sensor location. The charges collected are usually proportional to the irradiance but a non-linear mapping is applied before the image is written to the storage medium. The non-linear mapping is used to mimic the non-linearities in a film, to extend the dynamic range covered and also making the images more visually pleasing, since a linear mapping results in harsh images. The largest non-linearity is found at the saturation point where all intensities above the saturation value is mapped to the same value. This non-linear mapping is hard to know beforehand because it consists of a number of non-linear mappings inside the camera. This mapping is called the camera response curve or camera curve here denoted $f$, see Figure 1. This unknown camera response function infers a problem when generating HDR images. To generate HDR images we need to know the relative relationship between pixels from different images.

The transfer function $f$ also imposes a varying resolution in the mapping from irradiance to pixel values. The highest resolution and most reliable measures of true radiance are found for pixel values $Z_{ij}$ where $f$ has a steep slope. Therefore a window of reliable values is chosen, usually around the middle of the curve, and a weighted average over several exposures is used to improve the accuracy of the estimate of radiance in the final HDR image.

The camera response curve can be found directly from the input images, but this is a somewhat difficult optimization problem and will not be further discussed here.

\footnote{It should be mentioned that some digital imaging devices use other types of sensors, e.g. CMOS.}
Figure 1: **The diagram** shows a hypothetical nonlinear camera curve. The resolution in the mapping from irradiance $E$ to A/D-converted pixel values $Z$ varies along the curve and is best at the steep parts. Only a smaller range of irradiance values, denoted $A$, have good enough resolution to be used in the generation of HDR radiance maps.

Debevec et al. proposed a technique to find the camera response function $f(X)$. This technique is implemented in HDR-Shop.

The camera curve can also be estimated by direct measurement. One can photograph a target, usually a reflectance standard or other diffuse material, in a scene where the illumination or target reflectance can be changed in a controlled way. In the process of changing illumination, it is changed in steps from no illumination to bright light. For each illumination setting an image is captured with the camera. The amount of light reflected by the target is also measured using a photo-spectro meter or similar to get a true measurement. Capturing an image for all the settings and varying the illumination up to the camera saturation point will result in a set of point samples along the camera curve, i.e. each image corresponds to an irradiance value which was measured using the photo-spectro meter. One can then fit a curve through the points to find an estimation of the camera curve. This is usually done for each color channel in the camera, e.g. using red, green and blue illumination respectively.

### 2.2 High Dynamic Range Formats

Ordinary digital image formats like gif and jpeg etc. use 8 bits per color channel. This means that they can describe 256 different intensities per channel. The dynamic range is therefore in the order of two magnitudes $1:100$. This is sufficient for viewing the images on a computer display, but the true dynamic range of the radiance in the scene photographed is usually much bigger than that. For extreme scenes, the bright parts might be 100,000 to 1,000,000 times brighter than the dark areas. The solution is to use an image format with a higher number of bits per color channel and pixel. 32 bits per
Figure 2: An example of a weighting function used to weight out small pixel values and to remove saturated pixels.

color channel is enough to cover the dynamic range of the radiance. Such images are called High Dynamic Range Radiance Maps or HDR images. There are a number of such image formats, e.g. Greg Ward’s RGBE format, Pixar’s floating point tiff and the SGI LogLuv format etc. During this lab we will use the RGBE format.

2.3 Constructing HDR images

The technique for generating HDR images uses a set of images from the same imaging device with different exposures. That is a number of low dynamic range images with different exposure time are used. The choice of exposure times used depends on the scene. The shortest exposure time is chosen so that the brightest area in the scene is not saturated in the resulting image. The longest exposure time is chosen so that it can describe the darkest interesting area in the scene good enough for the purpose. An exposure $X_{ij}$ is defined as the irradiance $E_i$ at the sensor times the exposure time $\Delta t_j$.

\[
X_{ij} = E_i \Delta t_j
\]  

(1)

Where $i$ denotes the particular pixel or sensor location and $j$ the exposure.

As stated above the relative brightness values in the differently exposed images are usually not a true measurement of the radiance in the scene. The intrinsic non-linear mapping, the camera curve, $f(X)$ maps from irradiance seen by the sensor to pixel colors $Y_{ij}$. The pixel colors $Y_{ij}$ in the resulting image can be expressed as:

\[
Y_{ij} = f(X_{ij}) = f(E_i \Delta t_j)
\]  

(2)

The camera response function $f$ is assumed being monotonic and therefore also invertible, i.e. $f^{-1}$ exists. With a known camera curve the captured images can be
calibrated and the irradiance $E$ can be found

$$f^{-1}(Y_{ij}) = E_i \Delta t_j$$  \hspace{1cm} (3)

Small pixel values are sensitive to noise. Most cameras also saturates at a lower value than the maximum. Therefore a weighting function $g(Y_{ij})$, (see Figure 2), is applied to weight out and remove such values. The weighting is also used to values considered unreliable due to the varying resolution in the mapping from irradiance $E_i$ to A/D converted pixel values $Y_{ij}$. To avoid banding in the images a ramp is applied around end points of the window of reliable pixel values. This gives the following expression for the irradiance seen by the camera sensor at each pixel $(i)$:

$$E_{i,\text{weighted}} = g(Y_{ij}) \frac{f^{-1}(Y_{ij})}{\Delta t_j}$$  \hspace{1cm} (4)

The final HDR image is found by using the weighted mean of irradiance values from a number of exposures. For each pixel $Z_i$ in the resulting HDR image the weighted pixel values $Y_{ij}$ in the $N$ low dynamic range images with exposure time $\Delta t_j$ is averaged to produce the weighted mean output

$$Z_i = \frac{\sum_{j=0}^{N} (g(Y_{ij}) \frac{f^{-1}(Y_{ij})}{\Delta t_j})}{\sum_{j=0}^{N} g(Y_{ij})}$$  \hspace{1cm} (5)

3 Assignments

This section describes the two assignments for this practical. An assignment can be considered solved when you have a working .m file solving the assignment. Using .m files makes it easier for you and the assistant during examination of the lab. It also infers the nice possibility of reusing the functions you write.

All data needed for the lab can be found and downloaded from the practicals homepage. Create a directory for the lab and download and unzip the files needed.

3.1 The Camera Curve

The camera used for this lab is a Canon 10D. The camera response function curve was generated using HDRShop. The curve is used to map a pixel value in the low dynamic range images to the corresponding estimated irradiance value. The camera curve from HDRShop is the logarithm with base 2 of the irradiance value.

Set the working directory in Matlab to the one you created. Load the camera curve which is stored as a .m file by typing: `cameraCurve` at the Matlab prompt. Now you will have the curve loaded in a variable $C$.

Plot the camera curve, `plot(C)`, and the camera curve as $2^C$, `plot(2 .* C)`, and look at the result. The second plot shows the mapping from A/D converted pixel value to exposure denoted $X$ in Section 2.3.

Now open the file `calibrate.m`. As you see it contains a function with an empty body.
function l = calibrate(image, curve)
% The calibrate function calibrates the image, parameter 'image'
% using the camera curve, parameter 'curve'.
% 'curve' is the 3-by-256 element vector containing the response
% curve of the camera used.
%
% The return value from the routine is the calibrated image
l = image;

Write the routine for calibrating the input image using the camera curve. Run your routine on one of the images. Look at the result and compare it to the original image. Explain the differences.

3.2 Implement the HDR Algorithm

Open the file makehdr.m. It contains a skeleton for a routine that creates an HDR image from a set of low dynamic range images.

function hdr = makehdr(infile, min_exp, curve);
% MAKEHDR Creates a High-Dynamic Range Radiance Map
% from a set of low dynamic range images.
% 'infile' is a textfile containing the names of the
% low dynamic range images to use. The image names
% in the file should be ordered in ascending order
% with respect to the exposure times.
% 'min_exp' is the shortest exposure time used in the
% image sequence. 'min_exp' should correspond to the
% exposure time of the first image in the sequence.
% The exposures can be assumed being 1 f-stop apart,
% i.e. the exposure time is doubled between adjacent
% images in the sequence.
% 'curve' is the 3-by-256 element vector containing
% the response curve of the camera used.
% The return value hdr from the routine is the
% High Dynamic Range Radiance Map created from
% the image sequence.
hdr = zeros(512,512,3);

Write the routine that implements the HDR algorithm as described in Section 2.3. You should use the camera curve to calibrate the input images and apply a weighting function denoted $g$ in Section 2.3.

The shortest exposure time used in the image sequence is 1/4000 seconds and the images are approximately 1 f-stop\(^3\) apart.

Use the writehdr(img, filename, exposure) to write your HDR image to disk. This allows you to load it into HDRShop for viewing.

To look at your HDR image in Matlab you can use imshow() or imagesc(). Since Matlab is not an HDR viewer you will have to tonemap your HDR image before

\(^3\)1 f-stop is a factor of 2, e.g. $y = x \times 2^n$ gives that $y$ and $x$ are $n$ f-stops apart. In our case it means that the exposure time is doubled between the images in the sequence, $\Delta t_{j+1} = 2 \times \Delta t_j$. 
viewing. You can do this by taking the logarithm of the image, e.g. `imshow(log(hdr))`. You can also just normalize your image values and multiply them by some suitable factor `imshow(hdr/max(hdr(:))* k)`.

Now look at your image at different scalings:

```matlab
imshow(hdr/max(hdr(:)))
imshow(hdr/max(hdr(:)) * 10)
imshow(hdr/max(hdr(:)) * 20)
```

where `hdr` is your resulting high dynamic range image. Vary `k` to span the dynamic range of the image and explain the result.

Now when you know how the algorithms works, you can if you want use HDRShop to generate an High Dynamic Range radiance map from the sequence of images and compare to your own result. The camera curve can be loaded into HDRshop by choosing a custom camera curve.
4 Some Useful Matlab Functions

This section lists a number of functions that can be helpful during the practical. They are fully described in the manual pages in Matlab. Therefore they are not further explained here.

- `fopen(filename, mode);` opens the file 'filename' if it exists
- `fclose(FID);` closes the file pointed to by FID
- `frewind(FID);` sets the file pointer to the beginning of the file
- `fgetl(FID);` reads the current line from the file pointed to by FID
- `size(X);` returns the size of the parameter, can return more than one value
- `imread(fname);` reads the image fname
- `zeros(dim);` creates a matrix with zeros in all elements
- `reshape(X, dim);` reshapes X to a matrix according to dim