DIP PROJECT REPORT

Team: cut-copy-paste

Project Title: Artifact-free High Dynamic Range Imaging for Dynamic Scenes using Robust Patch-Based HDR Reconstruction.

Introduction:

High Dynamic Range Imaging is one of the most widely used techniques in the world of photography and image processing.

In image processing, high dynamic range imaging (HDRI) is a set of techniques that allows a greater dynamic range of exposures (the range of values between light and dark areas) than normal digital imaging techniques. The intention of HDRI is to accurately represent the wide range of intensity levels found in real scenes ranging from direct sunlight to shadows.

(from Wikipedia)

The first step in creating a high dynamic range (HDR) image is to capture a sequence of Low Dynamic Range (LDR) images. And then to put simply, when you aren't able to capture a single photo that includes detail in the darkest shadows and the brightest highlights of a scene, you'll want to capture multiple exposures and blend them together. This is where different softwares use various proposed and verified merging methods on the LDR stack images to produce the perfect HDR image. But **if the scene has motion content** ?. It creates various issues, like motion blur and unnecessary ghostly artefacts which creep into the final HDR image. So various researchers have published extensively in this field of research, and the paper that we chose for our project is one of the successful attempts at removing the ghostly artefacts from the image for dynamic scenes by improvising the state of the art techniques.

Example: Variants of Motion blur

Ghostly artefacts in the HDR images

Problem statement

The problem of removing motion artefacts for sequential HDR imaging has been the subject of extensive research. For pixels with motion, many algorithms use only a subset of exposures (in many cases only one) to produce a de-ghosted HDR. The fundamental problem with these techniques is that they cannot handle scenes with large motion if the changing portions of the scene have HDR content. So, the input sources are aligned to a reference exposure before merging them into an HDR image. The most successful algorithms use **optical flow (OF) to register the images**, but even these methods are still **brittle in cases of large motion** or complex occlusion/disocclusion. Since the "aligned" images produced by these algorithms often do not align to the reference very well, standard HDR merges of their results still have ghosting artefacts. Aligning the images to each other is a difficult problem that would be easier with information from the final HDR result which can be done using patch-based algorithms but the direct application of standard patch-based methods have not addressed the problem of HDR image reconstruction.

Motivation

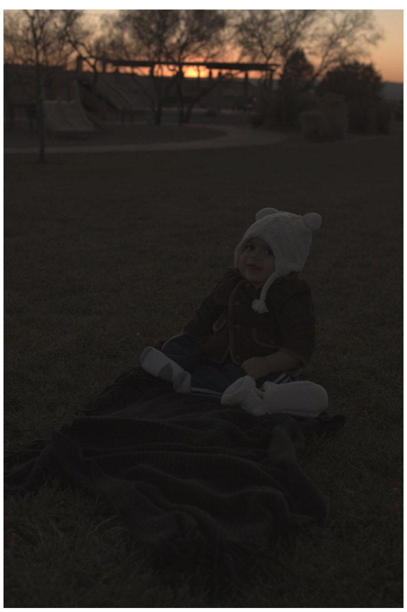
In practice, HDR imaging from a set of sequential exposures is a traditional way to capture high-quality images of static scenes. However, while fusing multiple images into a single radiance image, all the existent merging techniques severely suffer from artefacts caused by moving the camera and/or objects, because they assume a stationary scene. A camera motion causes global image transformation such as an affine or perspective transformations between different exposure images. If one takes photographs using a tripod, this problem might be reduced. **A more critical problem, however, is caused by an object motion which invokes inevitable ghost artefacts that make the same object appears multiple times in a generated HDRI**. Due to these reasons, practically it is a very important and **critical issue to produce a ghost-free HDRI from multiple images**. In this paper, a new HDRI generation method is proposed that is very effective in handling movements from multiple exposures.

Overview of the project [Input - Method - Output]

Input: We input a stack of low dynamic range images different in the exposures and different instances of capture.

Example Input :

LDR Image1
Exposure : 0.00625

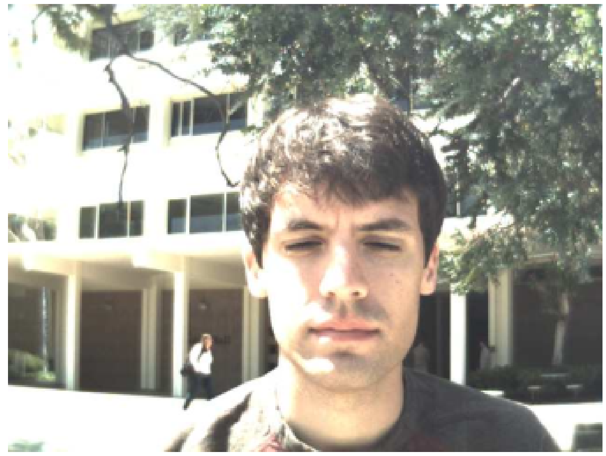


LDR Image2
Exposure : 1.000

LDR Image3
Exposure : 12.9960

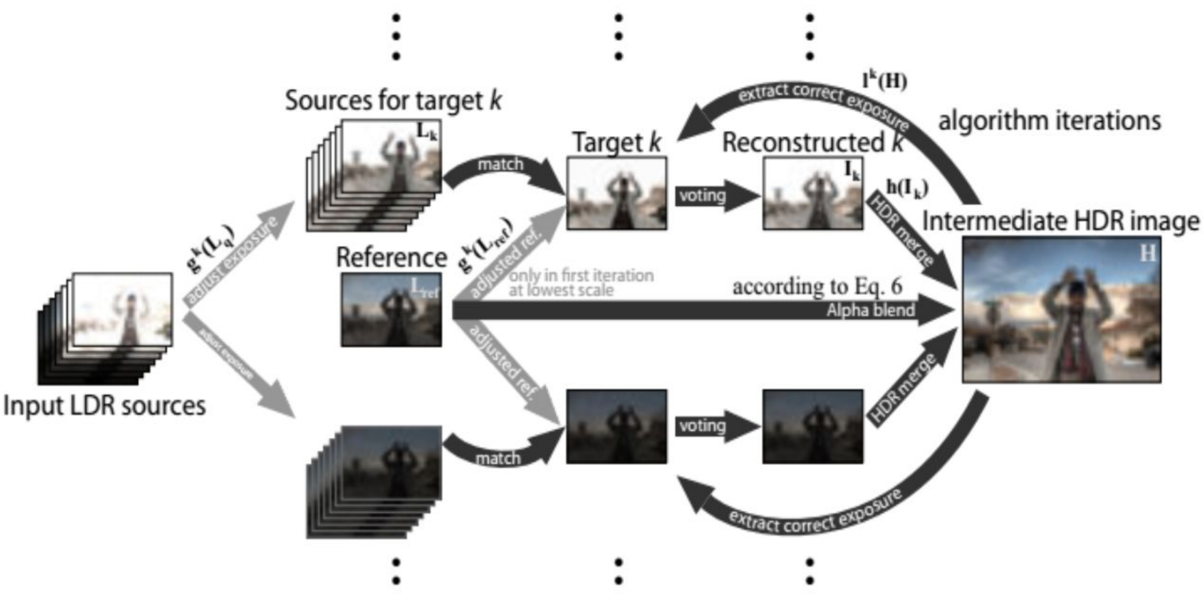
Short Exposure

Long Exposure



Method - Mathematical Formulation of the problem

Given a set of N LDR sources taken with different exposures and at different times (L1,...., LN - taking motion aspects into picture), our primary goal is to reconstruction HDR image H that is aligned to one of them (the reference, called Lref), but contains HDR information from all N exposures.



Operations performed

This paper uses a **patch-based algorithm**, which is based on a new HDR image synthesis equation,

HDR image synthesis equation does :

❑ Creating an HDR image containing information from all the exposures that are aligned with one of them, as if taken by an HDR camera at the same moment in time.

❑ And this turns into an **optimisation problem** , in which the optimal solution should match a reference image in the regions where it is well exposed, and in its poorly exposed regions are locally similar to the other LDR sources, containing as much information from them as possible.

(*The Figure in the bottom left corner contains 'Synthesis Equation'*)

❑ After optimising using the patch-based approach, we integrate the **alignment and merging processes**, instead of the conventional alignment and merging pipeline.

$$E(H, I_1, \ldots, I_N) = \sum_{p \in \text{pixels}} \Big[\alpha_{\text{ref}_{(p)}} \cdot \big(h(L_{\text{ref}})_{(p)} - H_{(p)}\big)^2 + (1 - \alpha_{\text{ref}_{(p)}}) \sum_{k=1}^{N} \text{MBDS}\big(I_k \mid g^k(L_1), \ldots, g^k(L_N)\big) + (1 - \alpha_{\text{ref}_{(p)}}) \sum_{k=1}^{N} \Lambda(I_{k_{(p)}}) \big(h(I_k)_{(p)} - H_{(p)}\big)^2 \Big].$$

Output: Final HDR image is outputted.

Approach Considered

> There are two possible approaches to this problem of removing the artefacts from the final HDR image, the first one is the **use of "de-ghosting" algorithms,** where they detect motion among different exposures but none of the approaches works accurately because they assume that pixel's radiance can be computed from the same pixel in all the exposures, but a moving HDR image has 'properly-exposed parts' in different parts of the image. So we switch to the second approach where the **LDR images are aligned to a reference image** before merging them to form a HDR image.

In this paper,

*They implemented a modified and improvised version of the current state of the art algorithm, proposed by *Zimmer et al*, where the optical flow algorithm is used to minimize the energy function consisting of gradient term. This approach doesn't actually synthesize new content but the approach here beats this method in that particular case by extension to multiple scales.

***Main Aim of the proposition**: Resultant HDR image, H is aligned with one of the input images, Lref but also contains the information from all the exposures.

***Terminology**

H	HDR image result which should look like L_{ref} but contain information from $L_1, \ldots, L_N$
$l^k(H)$	produces an LDR image at exposure k from HDR image H: $l^k(H) = \text{clip}\big((H/\text{exposure}(k))^{1/\gamma}\big)$
$h(L_k)$	maps LDR image L_k to the linear HDR radiance domain: $h(L_k) = (L_k)^\gamma \times \text{exposure}(k)$
$g^k(L_q)$	maps the q^{th} LDR source to the k^{th} LDR exposure: $g^k(L_q) = l^k(h(L_q))$

So, H must be very close to Lref, to help preserve as much information as possible from the well-exposed pixels of Lref. And to include all the information from the other exposures in where Lref is poorly exposed, in these parts H should be similar to any input source Lk. So there exists a similarity between H and Lref and H and Lk, this similarity metric is computed using, MBDS (Multi-source bidirectional similarity measure), MBDS is applied to all N source images in our input stack by defining an energy function that tries to keep each exposure n of the HDR image H as similar as possible to all input sources adjusted to that exposure.

$$E_{\mathrm{MBDS}}(H \mid L_1, \ldots, L_N) = \sum_{k=1}^{N} \mathrm{MBDS}\big(l^k(H) \mid g^k(L_1), \ldots, g^k(L_N)\big),$$

In this equation, MBDS is applied on H over all the possible input source images which ensure that every exposure of HDR image Lk(H) is similar to the adjusted version of all the N images in the regions where they are properly exposed.

$$E(H) = \sum_{p \in \mathrm{pixels}} \Big[\alpha_{\mathrm{ref}_{(p)}} \cdot \big(h(L_{\mathrm{ref}})_{(p)} - H_{(p)}\big)^2 + \big(1 - \alpha_{\mathrm{ref}_{(p)}}\big) \cdot E_{\mathrm{MBDS}}(H \mid L_1, \ldots, L_N)\Big].$$

This is the main synthesis equation, which has to be optimised. But it is difficult so, it has been reduced down to two simple optimisation equations, rather than one. To minimize this equation, an auxiliary variable **Ik** for lk(H) is introduced. Where, Ik is the LDR image that would be captured from the HDR image H by “exposing” it with the settings of the kth exposure.

Final Equation to be Optimised:

Here, the first term uses information from Lref wherever it is properly exposed and the second two terms fill in the poorly exposed regions with the information from the other exposures.

$$E(H,I_1,\ldots,I_N) = \sum_{p\in\text{pixels}} \Big[\alpha_{\text{ref}_{(p)}} \cdot (h(L_{\text{ref}})_{(p)} - H_{(p)})^2 +$$
$$(1-\alpha_{\text{ref}_{(p)}}) \sum_{k=1}^{N} \text{MBDS}\big(I_k \mid g^k(L_1),\ldots,g^k(L_N)\big) +$$
$$(1-\alpha_{\text{ref}_{(p)}}) \sum_{k=1}^{N} \Lambda(I_{k_{(p)}})(h(I_k)_{(p)} - H_{(p)})^2\Big].$$

Algorithm 1 Patch-based HDR image reconstruction algorithm

Input: unregistered LDR sources $L_1,\ldots,L_N$ and reference L_{ref}
Output: HDR image H, and "aligned" LDR images $I_1,\ldots,I_N$
Initialize: $\{I_1,\ldots,I_N\} \leftarrow \{g^1(L_{\text{ref}}),\ldots,g^N(L_{\text{ref}})\}$
for all scales s **do**
    for all optimization iterations **do**
        / Stage 1 – optimize for $I_1,\ldots,I_N$ in Eq. 4 */*
        for exposure $k = 1$ to $N, k \neq$ ref **do**
            $I_k \leftarrow \text{SearchVote}(I_k \mid g^k(L_1),\ldots,g^k(L_N))$
            $I_k \leftarrow \text{Blend}(I_k, l^k(H))$
        end for
        / Stage 2 – optimize for H in Eq. 4 */*
        $\tilde{H} \leftarrow \text{HDRmerge}(I_1,\ldots,I_N)$ [Eq. 5]
        $H \leftarrow \text{AlphaBlend}(h(L_{\text{ref}}), \tilde{H})$ [Eq. 6]
        / extract the new image targets for the next iteration */*
        $\{I_1,\ldots,I_N\} \leftarrow \{l^1(H),\ldots,l^N(H)\}$
    end for
end for
return H and $I_1,\ldots,I_N$

In the first stage, the algorithm minimizes for I1, . . . , IN. It first uses a search and vote which solves for the MBDS term by enforcing both the completeness and coherency terms. It then blends in lk(H) to each Ik using the previous H in order to encourage the solution to be close to the exposed value.

In Stage 2, the algorithm optimizes for the H variable in the first and last terms of the synthesis equation. First, it merges images I1, . . . , IN that were computed in the first stage into an intermediate HDR result H ˜.

$$\tilde{H}_{(p)} \leftarrow \frac{\sum_{k=1}^{N} \Lambda(I_{k_{(p)}}) h(I_k)_{(p)}}{\sum_{k=1}^{N} \Lambda(I_{k_{(p)}})}.$$

H ˜ contains information from all the other exposures which optimize for H. But still, it needs to be set to match the reference Lref in parts where Lref is well-exposed, to do this, our algorithm always injects the input reference directly into H using the appropriate alpha blending weights

$$H_{(p)} \leftarrow \alpha_{\text{ref}_{(p)}} \cdot h(L_{\text{ref}})_{(p)} + (1 - \alpha_{\text{ref}_{(p)}}) \cdot \tilde{H}_{(p)}.$$

This algorithm is performed at multiple scales, starting at the coarsest resolution and working to the finest.

Work Performed (Experiments, Software programming, etc.)

> The code is implemented in Matlab and has modularity for clearer understanding.

Flow of the code -

> Initial Step: The LDR stack images that are taken should be normalised accordingly for better results, in the paper, they have performed Gamma Normalisation for a **γ = 2.2,**

We can actually see for different values of gamma initialisation, output clarity in the dark parts actually varied.

<u>Results for varying gamma values</u>:

Gamma Value: 2.2

Gamma Value: 1.8

Linear Normalisation:

> And later on in the code, to extend it to multiple scales, we built pyramids for 10-scale images (has been reduced for less computation).These pyramids are later utilised in the optimisation done on a scale level.

> The next part of the algorithm takes care of the Search and Vote of the corresponding LDR image across all the other LDR images.

> We implement MBDS at this stage at which, for each target exposure level k, our method runs a dense search step a repeated number of times on all adjusted source exposures gk(Lq) using the current image at that level Ik as the MBDS target input.

$$H_{(p)} \leftarrow (1 - \alpha_{\mathrm{ref}_{(p)}})(\alpha^{+}_{(p)} \tilde{H}^{+}_{(p)} + \alpha^{-}_{(p)} \tilde{H}^{-}_{(p)}) + \alpha_{\mathrm{ref}_{(p)}} \cdot g(L_{\mathrm{ref}})_{(p)}$$

> This above equation is the basis for the merge operation that is performed in the later stage of the algorithm, H ˜ – which is a merge of the images that are lower than the

reference, and H ˜ + which is a merge of the images that are higher than the reference, the alpha values are calculated for each reference image, it is the per-pixel weighting, which indicated how well the reference image is exposed. It is calculated on the basis of the assumed values of Vmax and Vmin, initialised valid values of colour.

> The images are then later merged to form intermediate image HDR image H, and then lk(H) is used to produce the optimised exposed target images for the first stage in the next iteration.

[lk(H) produces an LDR image at exposure k from HDR image H : lk(H) = clip(H/exposure(k))^(1/ γ) *].*

This concludes the two stages of the algorithm but this algorithm is performed at multi-scale with the use of pyramid coding created by Lanczos downsampling methods and this improves the code more at a high-scale level and low -scale level for an image.

> We have implemented the algorithm on various datasets and also on our own dataset that we created using varying shutter speed and calculating the corresponding exposure times (it was difficult to find datasets of images without the accurate information about the exposure times).

Results (*experimented on the datasets that were available along with the corresponding exposure values)

Input: Above stack of LDR images,

Image-1: $\tfrac{1}{2}^6$

Image-2: $\tfrac{1}{2}^0$

Image-3: 2^2

Image-4: 2^4

Upon running the algorithm on the above set of LDR images,

Output -

Resultant HDR Image:

Resultant HDR Image:

Implementation on Own Dataset

Input Stack

*Result is distorted because the position of capture changed across images, there is motion and occlusion and to accommodate the unseen pixels in the reference image, the pixels must be reconstructed from the other LDR sources as there is no valid information there. (*one of the limitations of the paper)

Failure cases- Our Result

Software produced HDR tone mapped image

*LDR REFERENCE IMAGE (added gaussian noise using imnoise)

* FINAL HDR IMAGE

Analysis of results (*including failure cases)

Our Result HDR Tone mapped Image

Implementing the algorithm in the paper was computationally very expensive and very much memory critical in spite of pyramidal coding for higher accuracy. As you can notice the difference between the two images above, one on the left is produced by implementing the paper whereas the one on the right is produced by Software Tools like Adobe to produce HDR tone mapped image from the given set of LDR images. The most possible reason here is for a stack of 10 images, the computation became highly impossible to be run on local machines with a i5/i7 processor, so we reduced the size of the LDR stack by removing few exposures and ran the algorithm on a reduced set, so the exposures aren't well matched for most of the pixels. And also, in the case of failure case, the background is very clear in the tonemapped image whereas, the image produced by the algorithm has no such clarity as we have missed pixel content across various exposures for cutting of the dataset. And one major drawback with this algorithm that we spotted is that since the whole HDR image is based upon the reference image, so when we add noise to it, it is carried on to the HDR output image also. This can be corrected by denoising all the images before performing the algorithm. The outputs produced here by the implementation of the algorithm are a result of compromise between the efficiency and computation power.

Github Link

Matlab Code Link: HDR_DIP_PROJECT

Dataset Link: Image Scenes

Task assignment

S.No	Tasks	Team Member
1.	Data Set Collection (including own)	Sanjana Gunna
2.	Pyramid Coding for multi-scale	Sai Praneeth Chokkarapu
3.	MBDS Understanding and Implementation	Sai Praneeth Chokkarapu
4.	Search and Vote Algorithm	Sanjana Gunna
5.	HDR Merge	Sanjana Gunna
6.	Working out the Alpha values (pixel-wise for multi-scale)	Sai Praneeth Chokkarapu
7.	Documentation	Sanjana Gunna
8.	Analysis on other source images	Sai Praneeth Chokkarapu

Acknowledgement

We have borrowed few existing publicly-available implementations of standard HDR processing techniques from few published papers**.**

* MBDS Implementation - *Barnes et al.* [2009]

* Search and Vote Acceleration - *Patch Match Algorithm*

* Merge Approach - *Kang et al* [2003]

And we also used the datasets (with existing EXIF data for the exposures) mentioned above in the link.