Reconstructing HDR Images using Non-Learning and Deep Learning Based Multi-exposure Image Synthesis Techniques

Bin (Claire) Zhang, Megan Zhang
Department of Electrical Engineering, Stanford University

Motivation
- High dynamic range (HDR) imaging aims to present a greater range of luminance that is similar to that experienced by the human visual system. Its applications include VR, autonomous vehicles, and photography.
- Non-learning based HDR synthesis algorithms find the best weights to fuse multi-exposure low dynamic range (LDR) images. We adapted from Debevec's algorithm\(^1\) and utilized a different fusion technique for luminance and chroma channels respectively.
- Different learning models (CNN, GAN\(^2\), meta-learning\(^3\)) have been used for multi-exposure image synthesis. We selected U-Net as the backbone architecture and studied the effect of various training schemes (optimizer, loss, and regularization) on the image fusion results.

Related Work
- Debevec's algorithm\(^1\) fuses multiple LDR images by weighting more on pixels with values closer to the center of dynamic range at each exposure. Another hybrid fusion method\(^4\) calculates weights differently for the luminance and chroma channels. Both methods are limited in the sense that they require more information, such as exposure time, on the input images.
- Learning methods for LDR-HDR conversion require output reconstruction to full image size. To satisfy such requirement, U-Net is a good candidate model for its compact, symmetric encode-decoder style architecture. No additional information other than LDR input and HDR ground truth is needed.

References

New Technique

Non-learning based method
- Convert to YCoCr and extract the luminance (Y) channel
- HDR synthesis using Debevec's method
- Merge luminance (Y) channel
- U-Net

Deep learning based method
- U-Net
- HDR prediction
- Loss

Experimental Results

Comparison of HDR results
Debevec’s Algo Non-learning based Deep learning based

Comparison of PSNR (dB) for different objective functions and optimizers

<table>
<thead>
<tr>
<th></th>
<th>Adam</th>
<th>L2</th>
<th>CosineSim</th>
<th>A=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>EV-2</td>
<td>31.74</td>
<td>36.37</td>
<td>31.74</td>
<td>36.37</td>
</tr>
<tr>
<td>EV0</td>
<td>31.74</td>
<td>36.37</td>
<td>31.74</td>
<td>36.37</td>
</tr>
<tr>
<td>EV+2</td>
<td>31.74</td>
<td>36.37</td>
<td>31.74</td>
<td>36.37</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Adam</th>
<th>L2</th>
<th>CosineSim</th>
<th>A=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>TV</td>
<td>10.08</td>
<td>10.87</td>
<td>10.87</td>
<td>10.87</td>
</tr>
<tr>
<td>L1</td>
<td>16.92</td>
<td>17.70</td>
<td>17.70</td>
<td>17.70</td>
</tr>
<tr>
<td>L2</td>
<td>14.21</td>
<td>15.09</td>
<td>15.09</td>
<td>15.09</td>
</tr>
</tbody>
</table>

Comparison of PSNR (dB) for different objective functions and optimizers

Deep learning based method

Input | GT | Prediction | Adam+L2 | CosineSim A=1
---|---|---|---|-----|
EV-2 | | | PSNR=31.74dB SSIM=0.9331 | PSNR=36.37dB SSIM=0.9653
EV0  | | | | |
EV+2 | | | | |

Deep learning based method

Input | GT | Prediction | Adam+L2 | CosineSim A=1
---|---|---|---|-----|
TV  | | | PSNR=10.08dB SSIM=0.9331 | PSNR=16.92dB SSIM=0.9653
L1  | | | | |
L2  | | | | |