MAIA - MedicAl Imaging and Applications

Advanced Image Analysis

Lectures on Advanced Color Image Processing

B2 – Image Blending and HDR Imaging

May/2017

Adrian Galdran - Post-Doctoral Researcher
Biomedical Imaging Lab – INESC TEC Porto (Portugal)
http://bioimglab.inesctec.pt/

High-Dynamic Range Imaging

Figure 1: A series of five photographs. The exposure is increasing from left (1/1000 of a second) to right (1/4 of a second).

High-Dynamic Range Imaging

What is High Dynamic Range (HDR) imaging?

Most digital cameras and displays capture or display color images as 24-bits matrices. There are 8-bits per color channel pixel values are in 0 – 255 for each channel. In other words, a regular camera or a display has a **limited dynamic range**.

8-bits are barely enough to capture all the details of a real scene. So, the camera tries to estimate the lighting and **automatically** sets the exposure so that the most interesting aspect of the image has good dynamic range, and the parts that are too dark and too bright are clipped off to 0 and 255 respectively

High-Dynamic Range Imaging

What is High Dynamic Range (HDR) imaging?

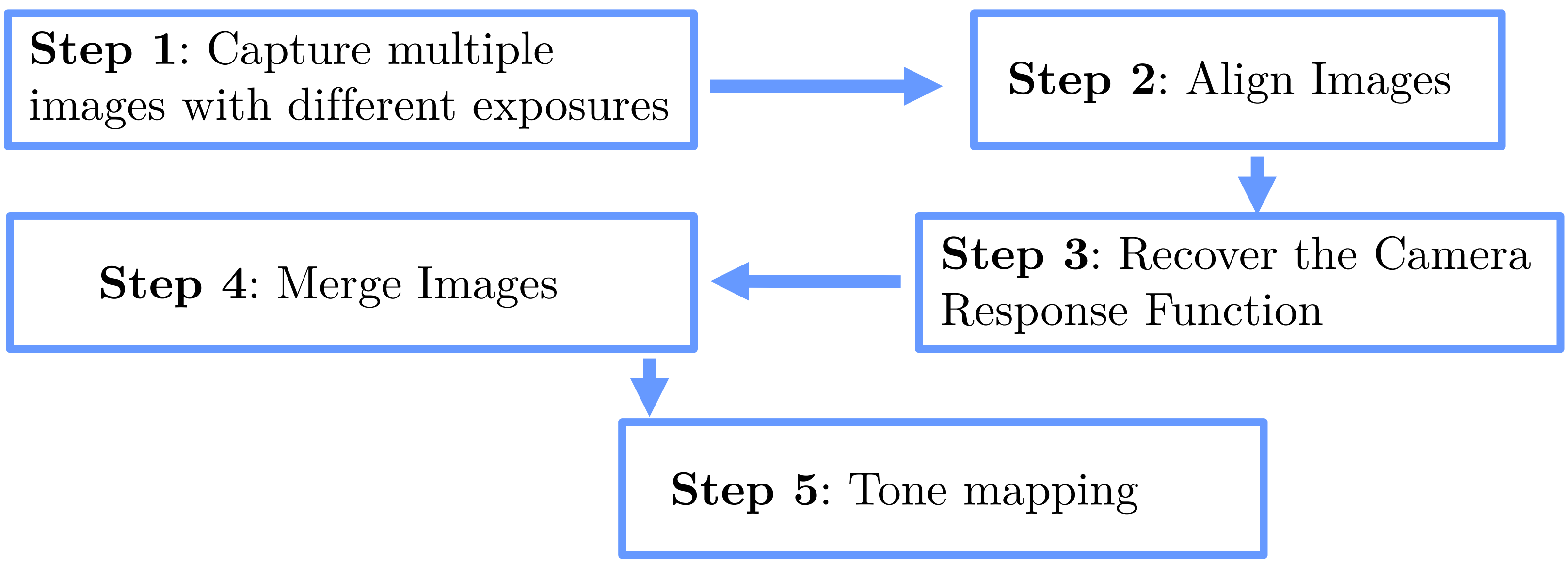
Normal Exposure

HDR Image

What is High Dynamic Range (HDR) imaging?

How does an iPhone capture an HDR image? It actually takes 3 images at **three different exposures**. The images are taken in quick succession so there is almost no movement between the three shots. The three images are then combined to produce the HDR image.

How does High Dynamic Range imaging actually work?



High-Dynamic Range Imaging

Step 1: Capture multiple images with different exposures

Most cameras have a feature called **Auto Exposure Bracketing** (AEB) that allows to take multiple pictures at different exposures with one press.

AEB acts quickly one after the other so the scene does not change. When we use HDR mode in an iPhone, it takes three pictures:

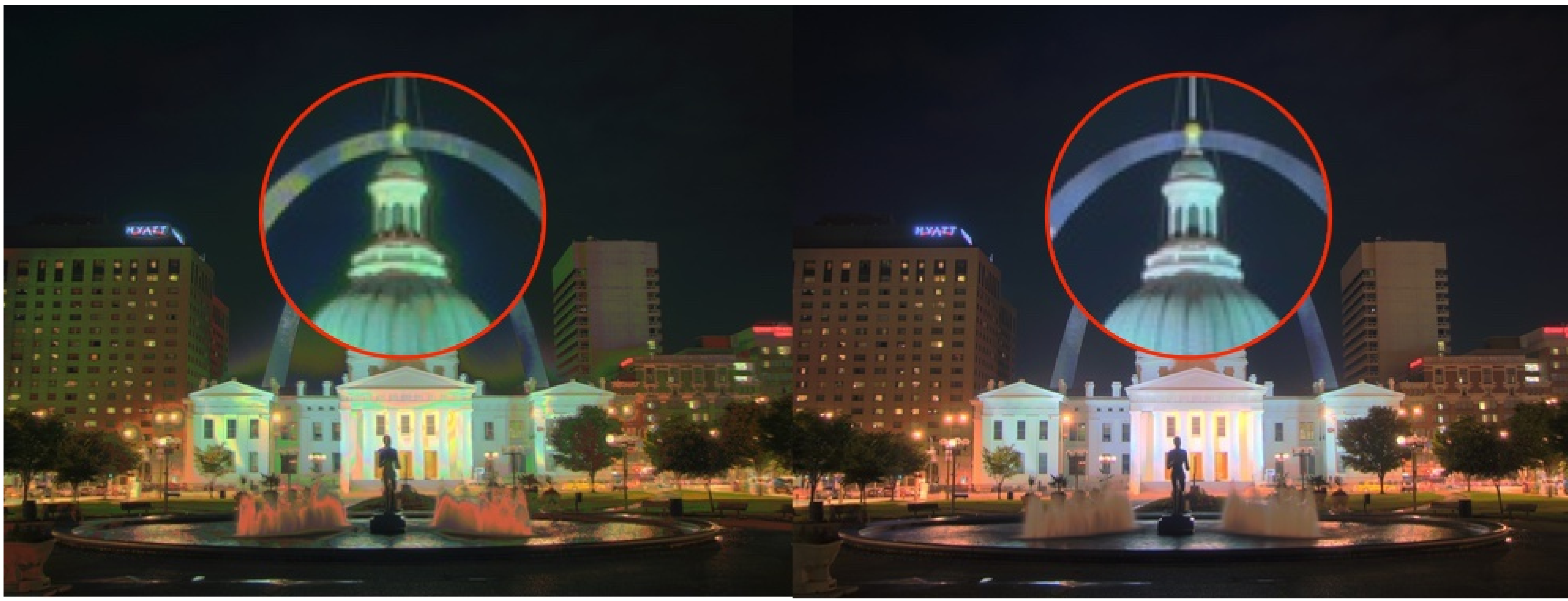
1) An **underexposed** image: This image is darker than the properly exposed image. The goal is the capture parts of the image that very bright.
2) A **properly exposed** image: This is the regular image the camera would have taken based on the illumination it has estimated.
3) An **overexposed** image: This image is brighter than the properly exposed image. The goal is the capture parts of the image that very dark.

However, if the dynamic range of the scene is very large, we can take more than three pictures to compose the HDR image

Example: 4 images, exposure time: 1/30, 0.25, 2.5 and 15 seconds

Step 2: Align Images

Misalignment of images used in composing the HDR image can result in ghosting artifacts:

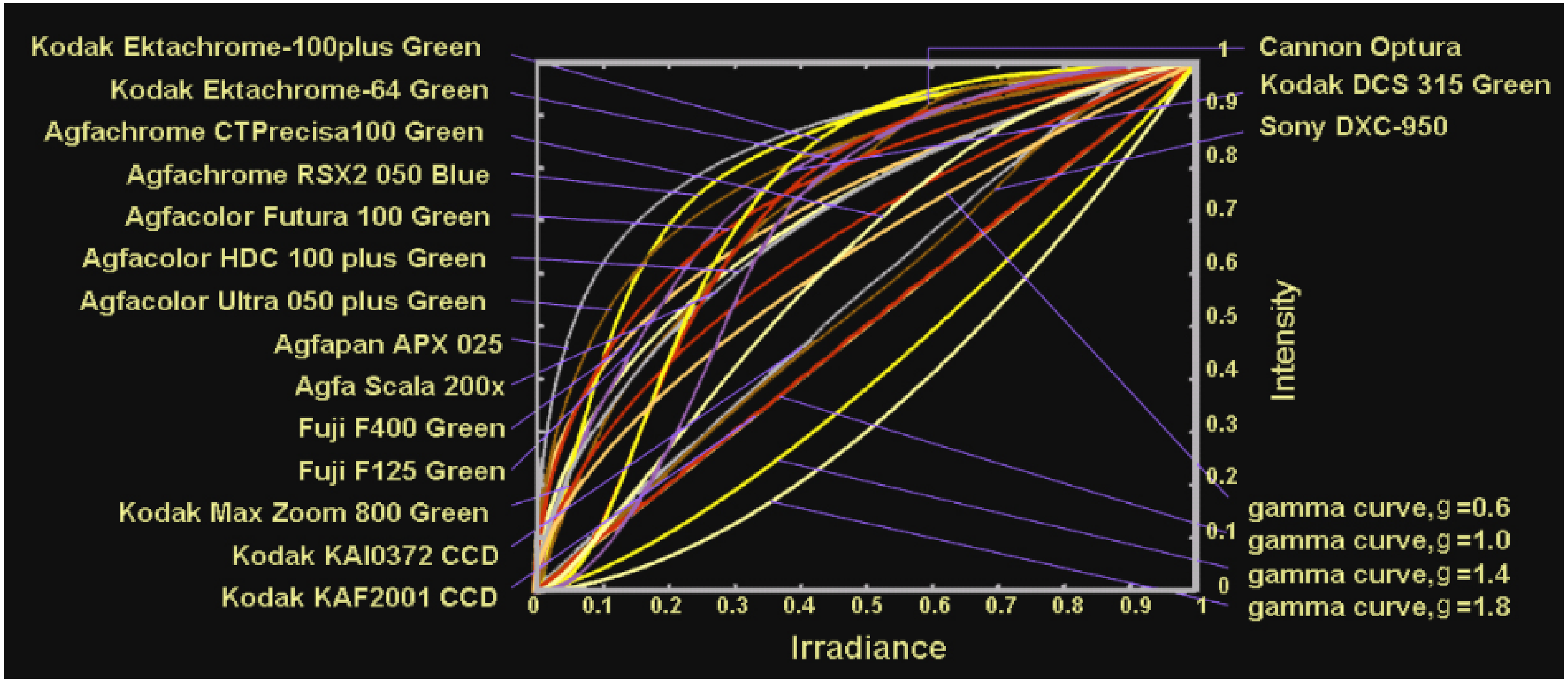


While creating an HDR image, professional photographers often mount the camera on a tripod. More on this on Thursday.

Step 3: Recover the Camera Response Function

The response of a typical camera is not linear to scene brightness.

What does that mean? Suppose, two objects are photographed by a camera and one of them is twice as bright as the other. When you measure the pixel intensities of the two objects in the photograph, the pixel values of the brighter object **will not be** twice that of the darker object.



Step 3: Recover the Camera Response Function

Without knowing or estimating the Camera Response Function (CRF), it is really hard to merge the images into one HDR image properly.

But, what do we exactly mean by merging multiple exposure images into an HDR image?

Consider a **single pixel** at some location. If the CRF was **linear**, the pixel value would be directly proportional to the exposure time unless the pixel is too dark (nearly 0) or too bright (nearly 255).

We can filter out these bad pixels (too dark or too bright), and estimate the brightness at a pixel by:

1) Dividing the pixel value by the exposure time.
2) Averaging this brightness value across all images where the pixel is not bad (too dark or too bright).

We can do this for all pixels and obtain a single image where all pixels are obtained by averaging "good" pixels.

Step 3: Recover the Camera Response Function

But the CRF is not linear and we need to make the image intensities linear before we can merge/average them by first estimating the CRF

But good news are, the CRF can be estimated from the images if we know the **exposure times** for each image.

Like many problems in computer vision, the problem of finding the CRF is set up as **an optimization problem** where the goal is to minimize an objective function consisting of a data term and a smoothness term. This results in a linear least squares problem which are solved using Singular Value Decomposition (SVD, see [1].

[1] Debevec & Malik, Recovering High Dynamic Range Radiance Maps from Photographs, Siggraph97

Step 4: Merge Images

We will speak about this later today.

Step 5: Tone Mapping

There is a final problem to solve. We've just built a HDR image, that has lots of intensities on it, i.e. there is a very large ratio between the brightest and darkest parts of the image.

An HDR image on a normal (low dynamic range) monitor will actually look very "sad":

Step 5: Tone Mapping

This is because that huge range of brightnesses has to be compressed to fit into a much smaller range of brightnesses. This results in an overall lack of contrast, hence the sadness/flatness.

Here the sky is very bright and the person much dimmer.

If we could use all the monitor's brightness range for the sky, it would look pretty good.

Step 5: Tone Mapping

This is because that huge range of brightnesses has to be compressed to fit into a much smaller range of brightnesses. This results in an overall lack of contrast, hence the sadness/flatness.

Here the sky is very bright and the person much dimmer.

If we could use all the monitor's brightness range for the sky, it would look pretty good.

But we'd totally lose the person. Likewise if you used all the monitors brightness range for the person it would look good, but we'd totally lose the sky.

Step 5: Tone Mapping

This is because that huge range of brightnesses has to be compressed to fit into a much smaller range of brightnesses. This results in an overall lack of contrast, hence the sadness/flatness.

Here the sky is very bright and the person much dimmer.

If we could use all the monitor's brightness range for the sky, it would look pretty good.

But we'd totally lose the person. Likewise if you used all the monitors brightness range for the person it would look good, but we'd totally lose the sky.

High-Dynamic Range Imaging

Step 5: Tone Mapping

It would be great if we could combine them in some way, or carefully distribute the brightness range we have so that we ca make most use of it. This is where **Tone Mapping** comes in.

What **Tone Mapping** does is instead of mapping the whole image into the monitor's brightness range in one go, it adjusts the contrast **locally** so that each region of the image uses the whole range for maximum contrast.

High-Dynamic Range Imaging

Step 5: Tone Mapping

You could apply any tone mapping technique to a non-HDR image.

In practice this is simple local contrast enhancement, e.g. CLAHE.

High-Dynamic Range Imaging

Example: 4 images, exposure time: 1/30, 0.25, 2.5 and 15 seconds




Let's see HDR in action! Open hdr-imaging.ipynb

Image Composition

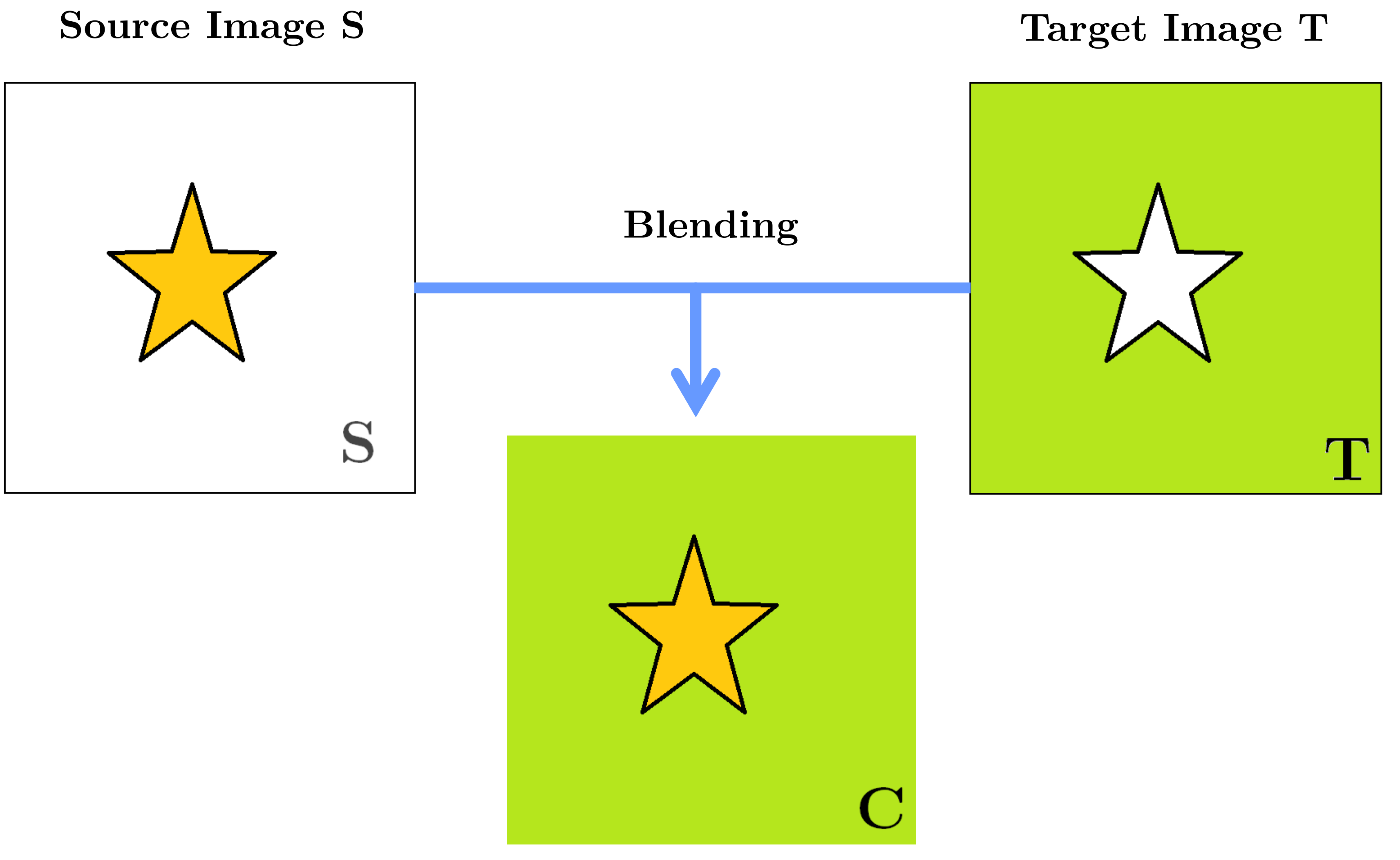
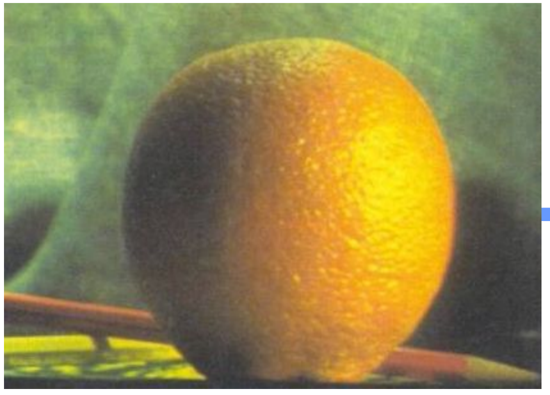


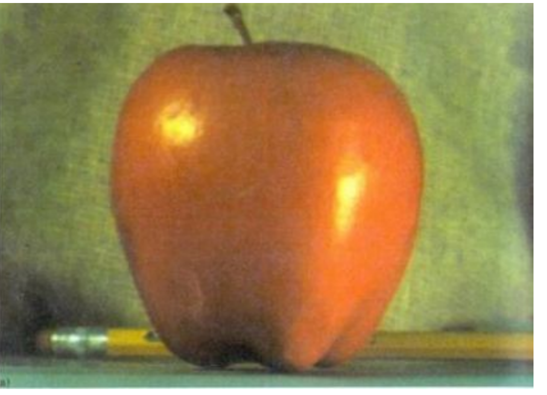
Image Composition

Source Image S



Blending

Target Image T



The Orangapple!

Image Composition

- **Naive Approach**: Given a binary mask **M**, compute **C = M · S + (1-M) · T**

  This gives bad results, since the seam between **S** and **T** will be visible

  How to **“hide”** that seam?

- **2nd Naive Approach**: Given a binary mask **M**, blur its edges and then blend:

$$\mathrm{C} = \mathbf{G}_\sigma * M \cdot \mathrm{S} + (1 - \mathbf{G}_\sigma * M) \cdot \mathrm{T}$$

- **Better Approach**: Multi-resolution blending with **Laplacian Pyramids**

  Merge different frequencies separately

532 IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-31, NO. 4, APRIL 1983

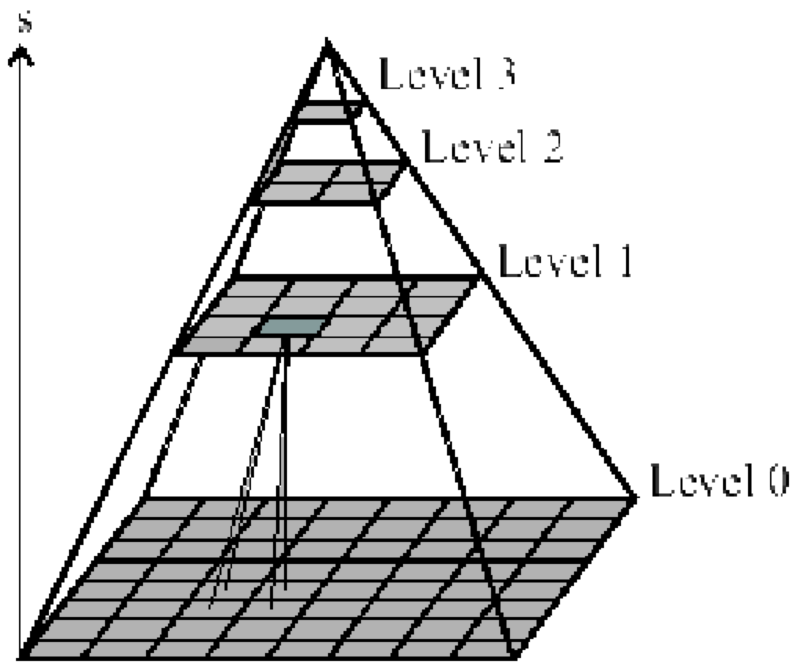
The Laplacian Pyramid as a Compact Image Code

PETER J. BURT, MEMBER, IEEE, AND EDWARD H. ADELSON

Image Pyramids

Blend Low-Frequency Regions Together

Blend High-Frequency Regions Together



Matlab: **impyramid**

Gaussian Pyramid

$$I_N = \text{downsample}(G_\sigma * I_{N-1})$$

$$\cdots$$

$$I_2 = \text{downsample}(G_\sigma * I_1)$$

$$I_1 = \text{downsample}(G_\sigma * I_0)$$

$$I_0 = I$$

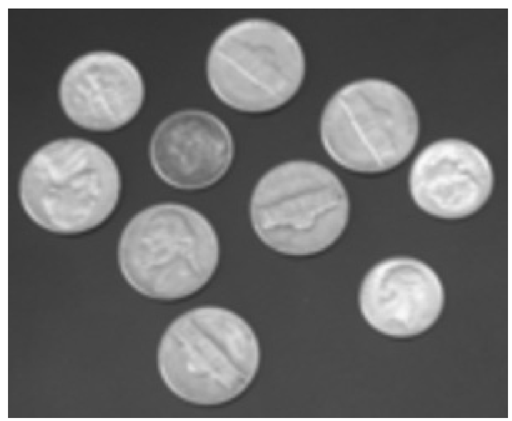
Image Pyramids

Blend Low-Frequency Regions Together

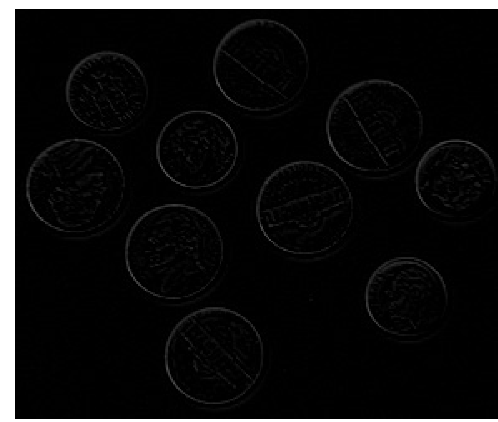
Blend High-Frequency Regions Together

Laplacian Pyramid

$-$



$=$

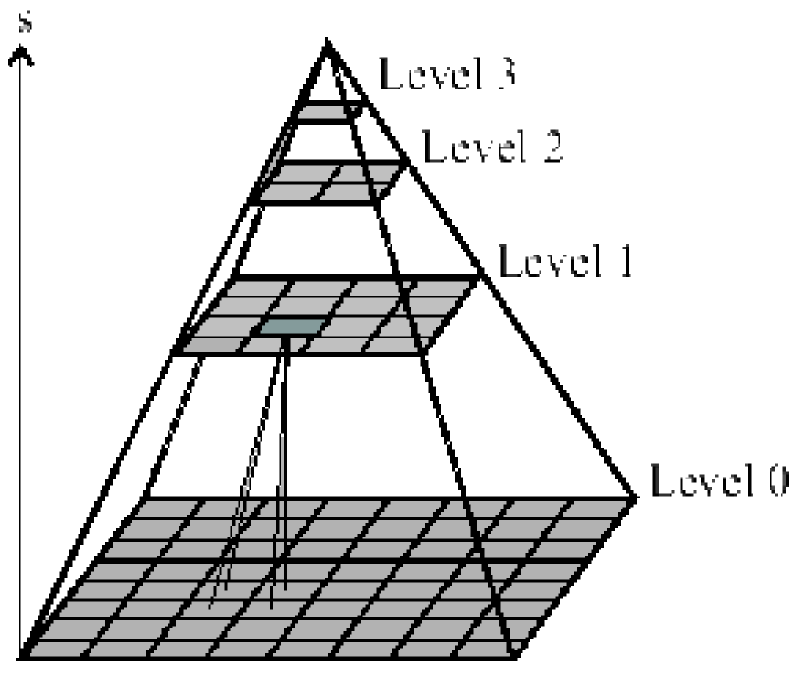


I_1 $\qquad$ $G_\sigma * I_1$ $\qquad$ L_1

Image Pyramids

Blend Low-Frequency Regions Together

Blend High-Frequency Regions Together



Laplacian Pyramid

$$L_n = I_n$$

$$\dots$$

$$L_2 = I_2 - G_\sigma * I_2$$

$$L_1 = I_1 - G_\sigma * I_1$$

$$L_0 = I - G_\sigma * I_0$$

Image Reconstruction from Pyramid

Reconstruction : From the Laplacian decomposition, you come back to the original image by upsampling and adding back edges iteratively to the smallest image:

$$I = \sum_{i=1}^{N} L_i \nearrow^2, \qquad L_N = I_N \qquad \textbf{(1)}$$

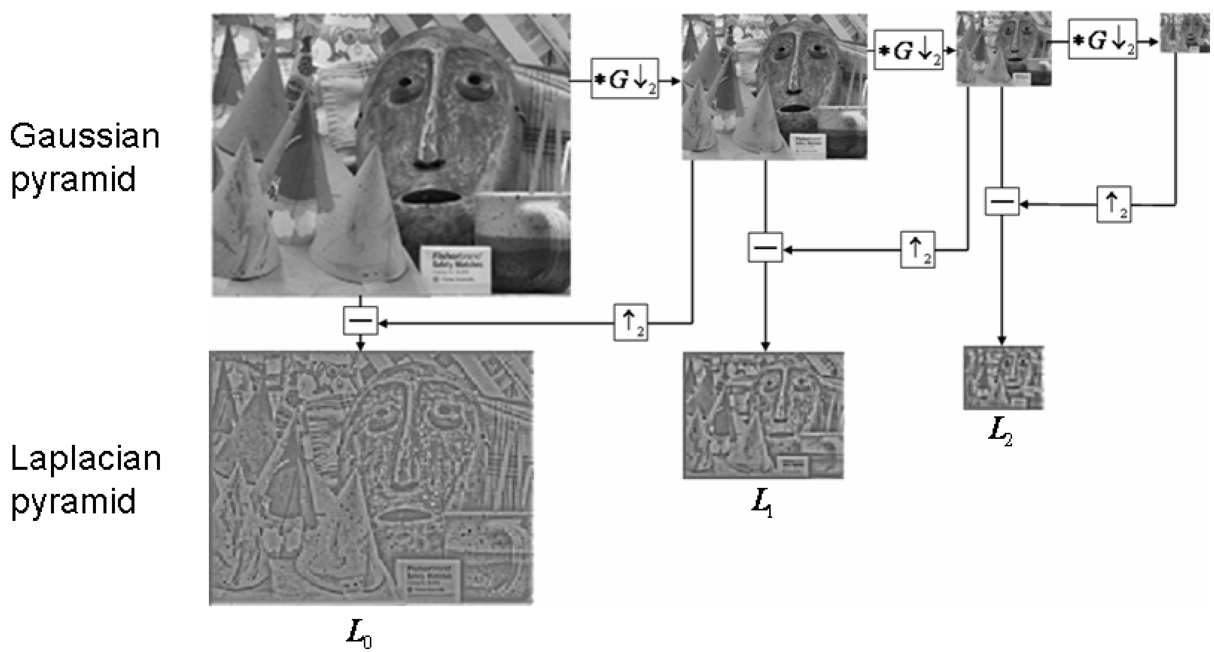


Image Reconstruction from Pyramid

Reconstruction : From the Laplacian decomposition, you come back to the original image by upsampling and adding back edges iteratively to the smallest image:

$$\boxed{I = \sum_{i=1}^{N} L_i \nearrow^2, \qquad L_N = I_N \qquad \textbf{(1)}}$$

Laplacian Image Blending

- We compute a **Gaussian Pyramid** for the mask, G_i^M
- We compute a **Laplacian Pyramid** for the source and target images, L_i^S, L_i^T
- We **blend at each level**: $I_i = G_i^M \cdot L_i^S + (1 - G_i^M) \cdot L_i^T, \quad i = 1, ..., N$
- We apply formula **(1)** to the multi-level blendings, $$I = \sum_{i=1}^{N} I_i \nearrow^2$$

Image Composition

Let's see simple Image Composition in action!
Open image-composition.ipynb

Image Matting

Let's see Image Matting in action!
Open alpha-matting.ipynb

Image Matting

Image Matting

Image Matting

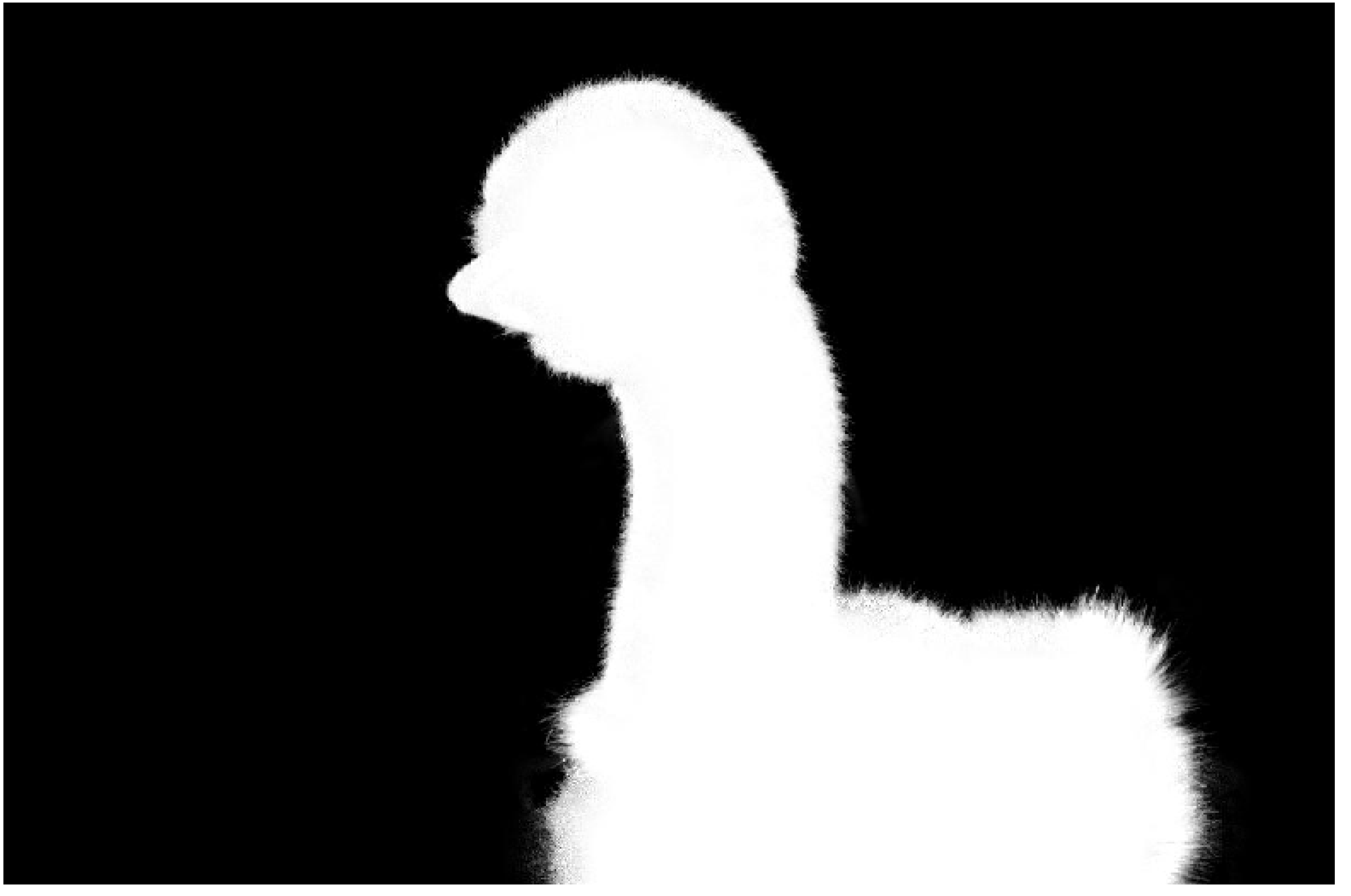


Image Matting

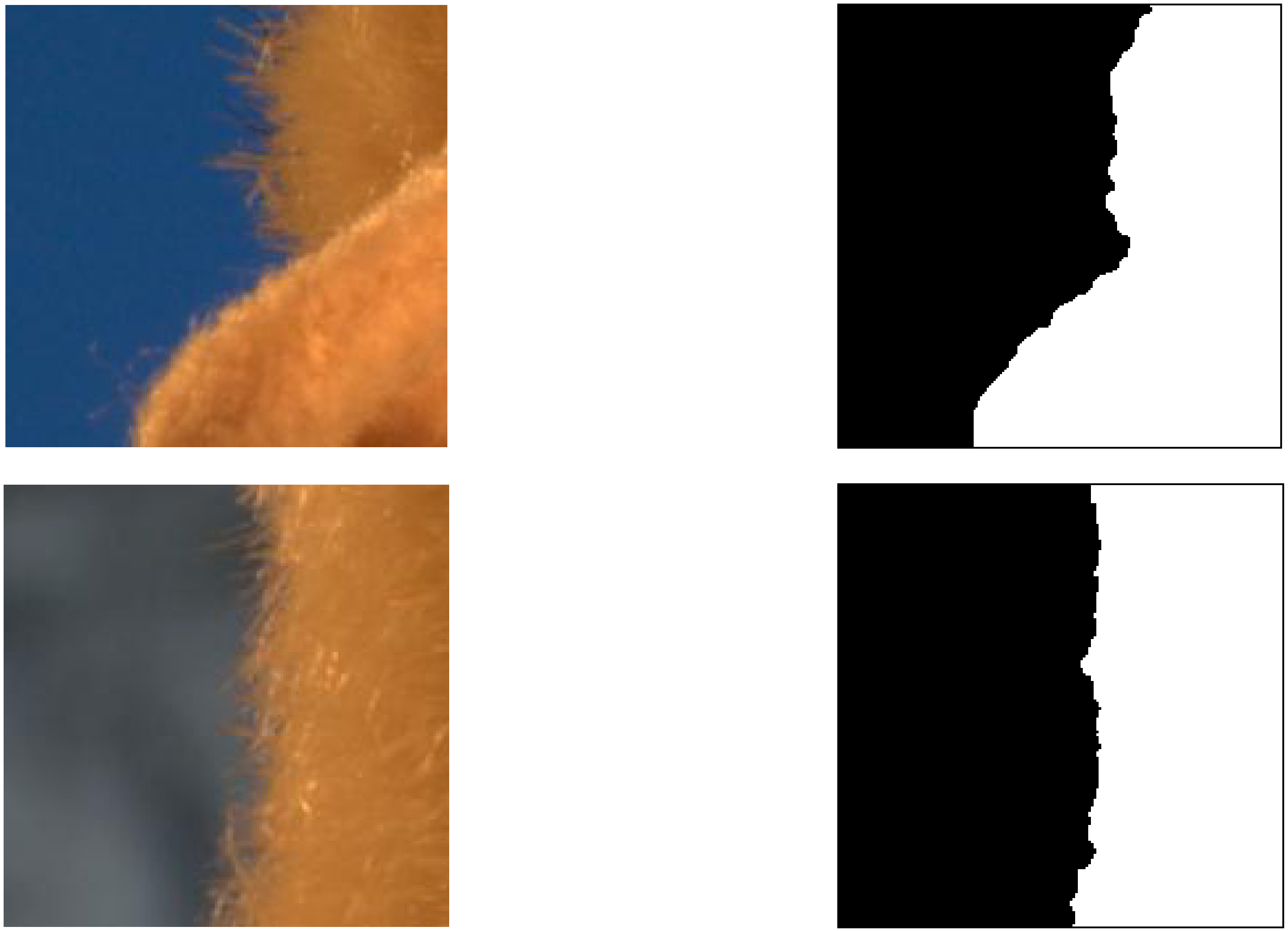
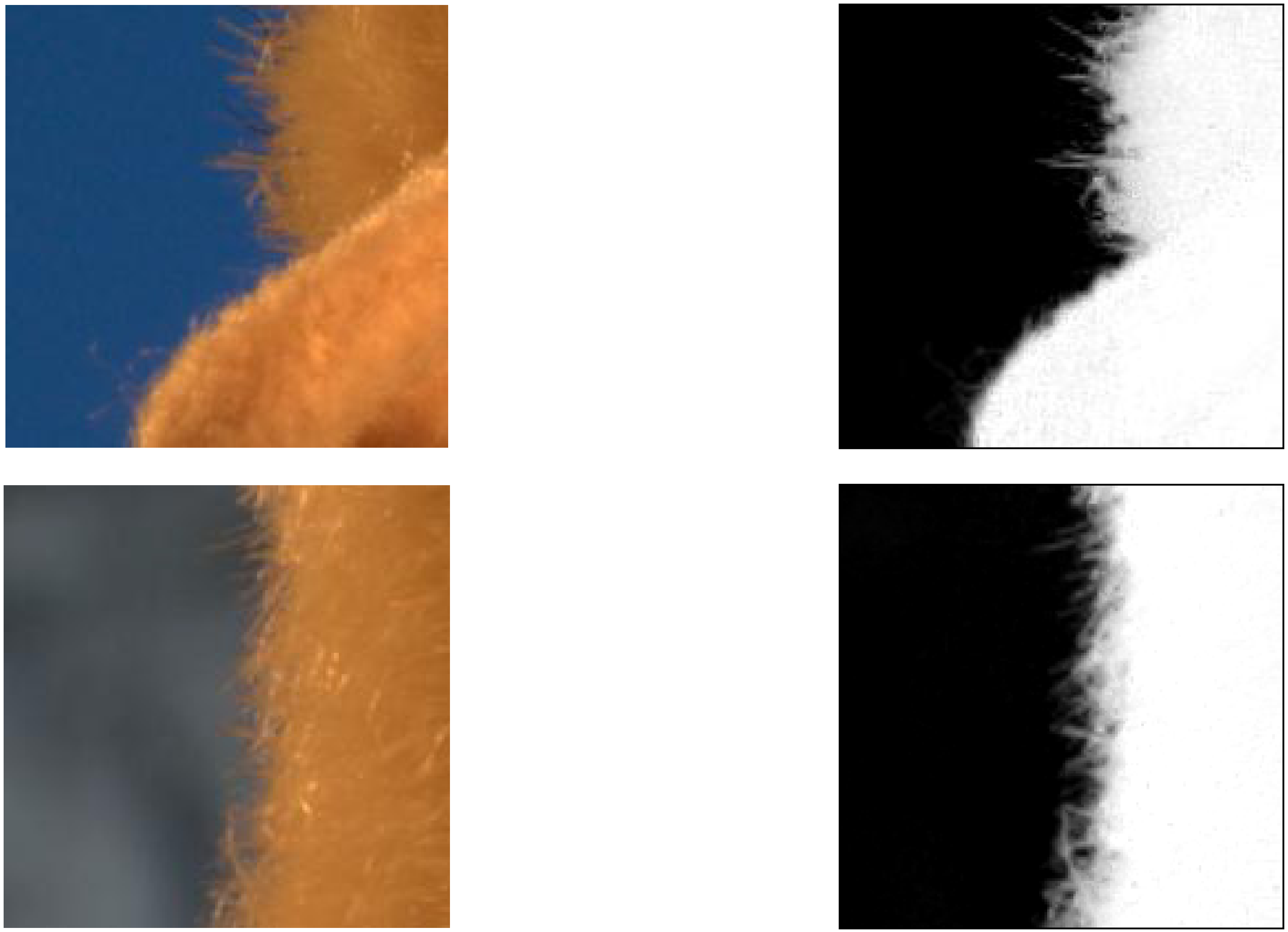


Image Matting



Multi-Exposure Image Fusion

Figure 1: A series of five photographs. The exposure is increasing from left (1/1000 of a second) to right (1/4 of a second).

HDR imaging requires to know the precise exposure time. The HDR image itself looks dark and is not pretty to look at. The minimum intensity in an HDR image is 0, but theoretically, there is no maximum. So we need to map its values between 0 and 255 so we can display it (**tone mapping**).

As you can see, assembling an HDR image and then doing tone mapping is quite tricky. Can't we just use the multiple images and create a tone mapped image **without ever going to HDR?**

The answer is yes, using **Exposure Fusion**.

Multi-Exposure Image Fusion

Multiple Exposure Image Composition

Figure 1: A series of five photographs. The exposure is increasing from left (1/1000 of a second) to right (1/4 of a second).

Exposure Fusion

Tom Mertens[1] Jan Kautz[2] Frank Van Reeth[1]

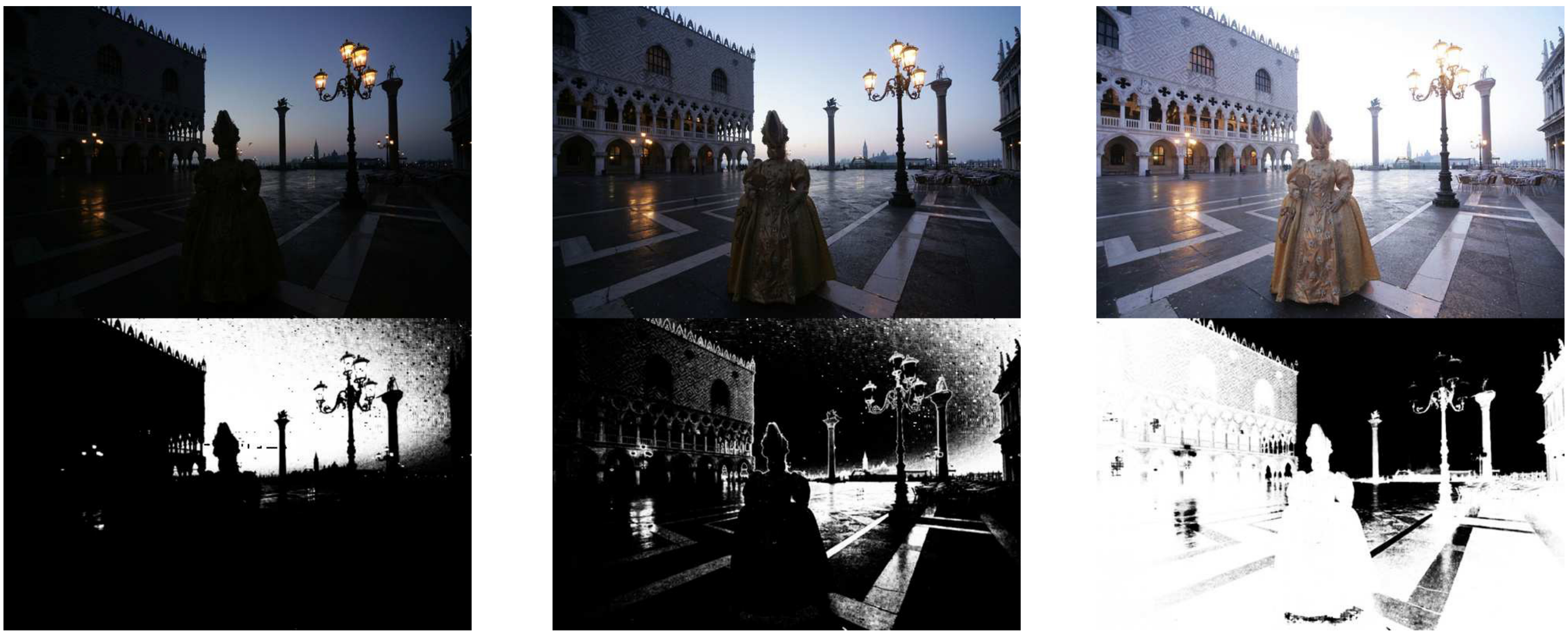
[1]Hasselt University — EDM
transationale Universiteit Limburg
Belgium

[2]University College London
UK

Multiple Exposure Image Composition

$$C = W_1 \cdot I_1 + 1 - W_2 \cdot I_2 \longrightarrow C = W_1 \cdot I_1 + W_2 \cdot I_2 + \ldots W_n \cdot I_n$$

$$\sum_{i=1}^{n} W_i = 1$$



Multiple Exposure Image Composition

$$C = W_1 \cdot I_1 + 1 - W_2 \cdot I_2 \longrightarrow C = W_1 \cdot I_1 + W_2 \cdot I_2 + \ldots W_n \cdot I_n$$

$$\sum_{i=1}^{n} W_i = 1$$

Multiple Exposure Image Composition

- How do we compute W_i?

$$\boxed{C = W_1 \cdot I_1 + W_2 \cdot I_2 + \ldots W_n \cdot I_n}$$

Contrast:
Absolute Value of Laplacian Filter Response, $W^C = \mathcal{L}(I)$ $\mathcal{L} =$

0	1	0
1	−4	1
0	1	0

Saturation:
Standard Deviation between R, G, and B values in each pixel, $\mu = \frac{I_R(i)+I_G(i)+I_B(i)}{3}$

$$W^{sat}(i) = \sqrt{((I_R(i)-\mu)^2 + (I_G(i)-\mu)^2 + (I_B(i)-\mu)^2)/3}$$

Exposure:
Deviation from gray value, $W^{exp}(i) = \exp\left(-\frac{(I(i)-0.5)^2}{\sigma^2}\right)$

$$\boxed{W = W^C \cdot W^{sat} \cdot W^{exp}} \Rightarrow \boxed{W_j = W_j^C \cdot W_j^{sat} \cdot W_j^{exp}} \quad \boxed{\sum_{j=1}^n W_j = 1}$$

Let's see Mulitple Exposure Image Fusion in action!
Open mef.m files
Open mef.ipynb