05_DVC4IL: Digital Imaging / Visual Computing
Lab Report 2

Eren Yeșiltepe

May 22, 2024

Abstract

In this lab unit, we try to implement hybrid images. Hybrid images are based on the multiscale processing of images by the human visual system and are motivated by masking studies in visual perception. These images can be used to create compelling displays in which the image appears to change as the viewing distance changes.

2.1 Task

The goal of this assignment is to create hybrid images. Hybrid images are static images that change in interpretation as a function of the viewing distance. The basic idea is that high frequency tends to dominate perception when viewed from a close distance but, at a distance, only the low frequency (smooth) part of the signal can be seen. By blending the high frequency portion of one image with the low-frequency portion of another, you get a hybrid image that leads to different interpretations at different distances.

In this trial, Gaussian filters were employed as recommended by the literature. Specifically, a standard Gaussian filter was utilized for the low-pass filtering process. For high-pass filtering, I implemented a method involving the original image subtracted by its Gaussian-blurred version, specifically img - cv2.GaussianBlur(img, (0,0), sigma) + 160. This approach effectively highlighted the high-frequency components of the image. Subsequently, the images were merged using alpha blending. Prior to blending, I ensured that the images were resized to identical dimensions to facilitate seamless integration.

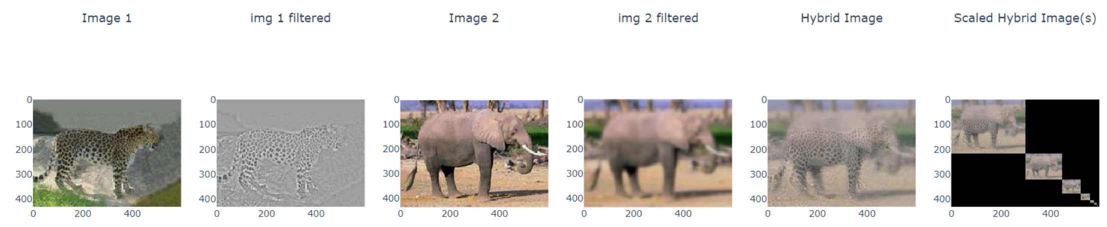
2.1.1 Including code

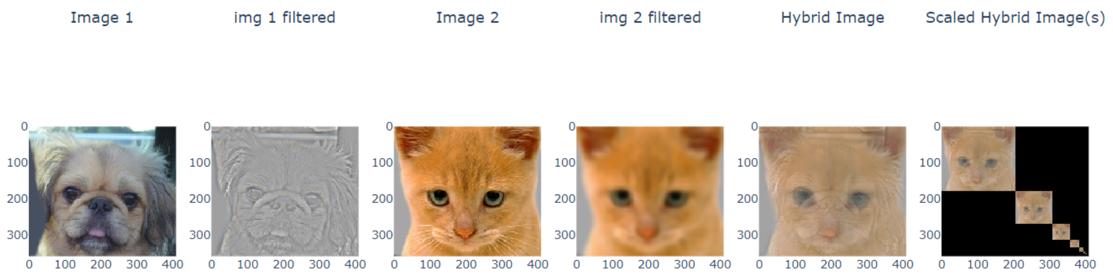
```
# image 1 and 2
img1 = cv2.imread("hybrid_images/cheetah.png")
img2 = cv2.imread("hybrid_images/elephant.png")
assert img1 is not None and img2 is not None, "Image not found"

def highpass(img, sigma):
    return img - cv2.GaussianBlur(img, (0,0), sigma) + 160
```

```python
def normalize_image(image):
    normalized = cv2.normalize(image, None, 0, 255, cv2.NORM_MINMAX)
    return np.uint8(normalized)

def equalize_images(img1, img2):
    # Find the minimum dimensions
    min_height = min(img1.shape[0], img2.shape[0])
    min_width = min(img1.shape[1], img2.shape[1])

    # Crop the images to the minimum dimensions
    img1_cropped = img1[:min_height, :min_width]
    img2_cropped = img2[:min_height, :min_width]

    return img1_cropped, img2_cropped

def alpha_blend(img1, img2, alpha):
    """
    Blend two images using an alpha value.

    Parameters:
    - img1: First input image.
    - img2: Second input image.
    - alpha: Alpha value for blending (0.0 to 1.0).

    Returns:
    - Blended image.
    """
    img1,img2 = equalize_images(img1,img2)
    # Ensure the alpha value is within the valid range
    alpha = np.clip(alpha, 0.0, 1.0)

    # Compute the beta value
    beta = 1.0 - alpha

    # Perform the blending
    blended = cv2.addWeighted(img1, alpha, img2, beta, 0.0)

    return blended


img1_filtered = highpass(img1,6)
img2_filtered = cv2.GaussianBlur(img2, (0,0), 6)
hybrid = alpha_blend(img2_filtered,img1_filtered,0.5)

# display the 2 input images and the hybrid image
fig = make_subplots(rows=1, cols=6, subplot_titles=("Image 1","img 1 filtered", "
    Image 2","img 2 filtered", "Hybrid Image", "Scaled Hybrid Image(s)"))
fig.add_trace(go.Image(z=img1[:,:,::-1]), row=1, col=1)
fig.add_trace(go.Image(z=img1_filtered[:,:,::-1]), row=1, col=2)
fig.add_trace(go.Image(z=img2[:,:,::-1]), row=1, col=3)
fig.add_trace(go.Image(z=img2_filtered[:,:,::-1]), row=1, col=4)
fig.add_trace(go.Image(z=hybrid[:,:,::-1]), row=1, col=5)
small_img = multi_scale_image(hybrid)
fig.add_trace(go.Image(z=small_img[:,:,::-1]), row=1, col=6)
fig.show()
```

Colab URL: “https:colab.research.google.comdrive1plRwp$_{A}OfDxwpM-kwSWdFBKQpvLPwXW2?$
$sharing$”

2.1.2 Results





As it can be clearly seen in the examples, the results of this trial (specifically the elephant and the cheetah example) gave satisfactory results.

Summary and comments

For me the biggest challenge of this assignment was to find the correct blending. I figured out via blending the filtered photos in a real image manipulation tool. Overall it was quite a fun assignment.