Data-driven Hallucination

Team: Image Processors
Manish -2018101073
Akshat Goyal - 2018101075
Dixit Kumar Garg - 2018101077
Khadiravana - 2019121008

Original Paper link

Problem statement

Given a single image as input, we want to automatically create a plausible-looking photo that appears as though it was taken at a different time of the day. This should be done using a fixed database of time lapse videos, of which the input image may not be a part of. Also we want to make the image look realistic, while preserving the colour schema.

Input image at “blue hour”

Hallucinations at night

Utility

Time of day and lighting conditions are critical for outdoor photography. Photographers spend much effort getting to the right place at the perfect time of day, going as far as dangerously hiking in the dark because they want to reach a summit for sunrise or because they can come back only after sunset. In addition to the famous golden or magical hour corresponding to sunset or sunrise , the less-known “blue hour” can be even more challenging because it takes place after the sun has set or before it rises and actually only lasts a fraction of an hour when the remaining light scattered by the atmosphere takes a deep blue color and its intensity matches that of artificial lights. Most photographers cannot be at the right place at the perfect time and end up taking photos in the middle of the day when lighting is harsh. In this project, we implement an automatic technique that takes a single outdoor photo as input and seeks to hallucinate an image of the same scene taken at a different time of day.

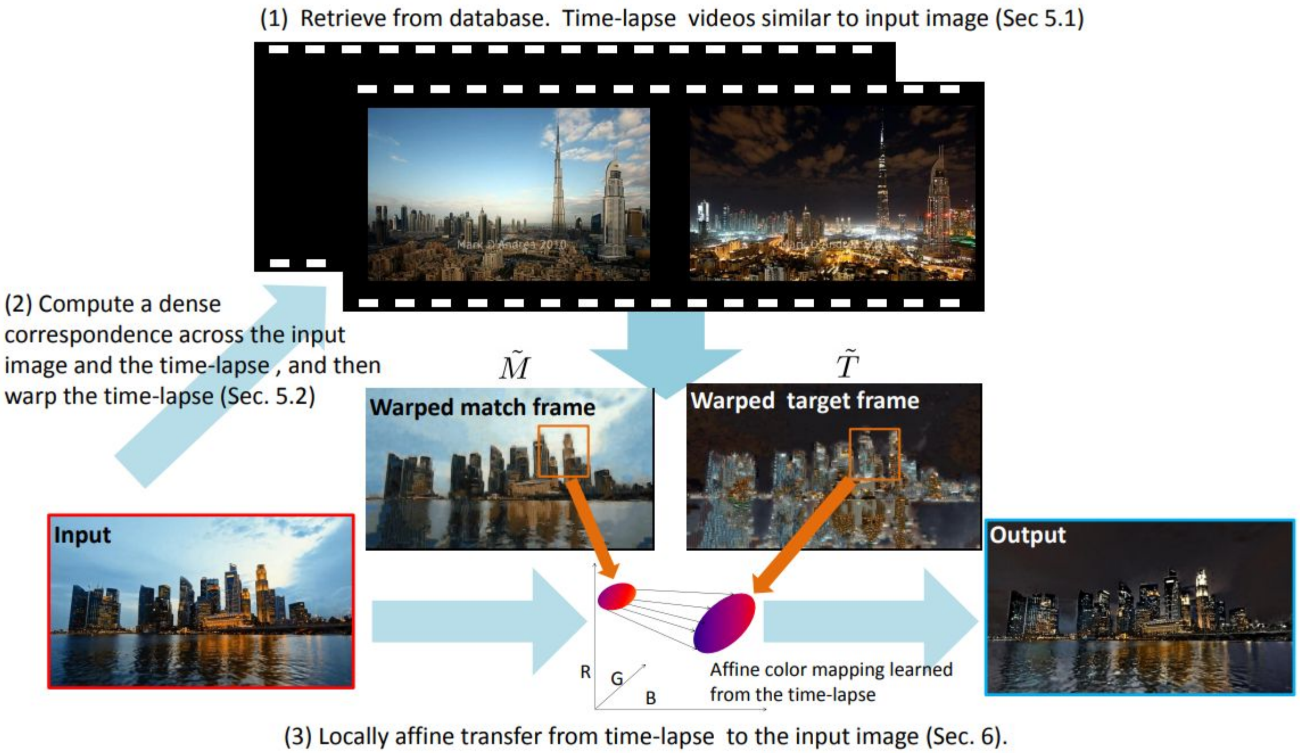
Golden Hour

Blue Hour

Milestones

1. Reading and understanding the paper
2. Global Matching
3. Frame Selection
4. Local Matching
5. Locally Affine color transfer
6. Denoising image

1. Reading and understanding the paper



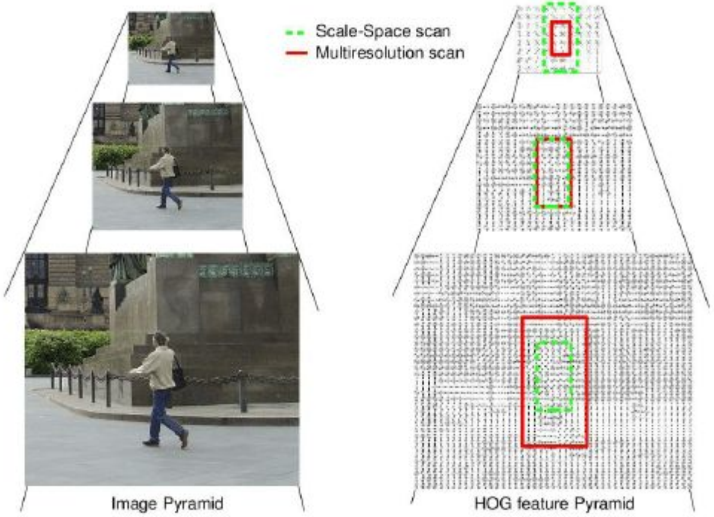
2. Global Matching

The first step of our algorithm is to identify the videos showing a scene similar to the given input image. We employ a standard scene matching technique in computer vision. We sample 5 regularly spaced frames from each video, and then compare the input to all these sampled frames.To assign a score to each time-lapse video, we use the highest similarity score in feature space of its sampled frames. We used Histograms of Oriented Gradients (HOG) [Dalal and Triggs 2005] for getting features.

Pyramid-HOG Algorithm

For selecting best videos(global matching) and selecting the best frame , we need to compare images. To reduce computational cost we need a compact representation which is provided by Pyramid Histogram of Oriented Gradient (PHOG).

The objective of the PHOG is to take the spatial property of the local shape into account while representing an image by HOG. The spatial information is represented by tiling the image into regions at multiple resolutions based on spatial pyramid matching .



3. Frame-Selection

Now that we have a set of matching videos, for each of them, we seek to retrieve a frame that matches the time of day of the input image. We use the color histogram and L2 norm to pick the matched frame.

Let H() be the histogram function. I be the image and M be the matched frame and X be the set of frames taken from the matched video at equal intervals then,

$$H(i) = [I_o, I_1, I_2, \ldots, I_{255}]$$
$$cost(x) = \sum_i (I_i - X_i)^2$$
$$M = X \ni (cost(X) = \min_{\forall y \in X} cost(y))$$

Our Results

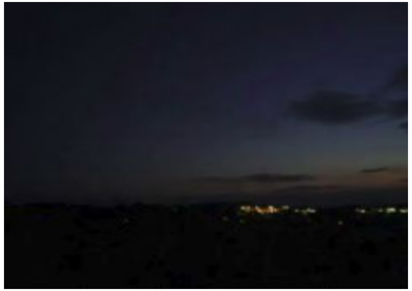
Input Image

Matched Frame

Input Image

Matched Frame

Input Image



Matched Frame

Milestones Remaining

1. Local Matching

   We seek to pair each pixel in the input image I with a pixel in the matched frame. For this, we formulate the problem as a Markov random field (MRF).
2. Locally Affine Color Transfer
   a. We want it to explain the color variations observed in the timelapse video.
   b. We want a result that has the same structure as the input and that exhibits the same color change as seen in the time-lapse video.
3. Denoising image

   Input The affine mapping has a side effect that it may magnify the noise existing at the input image, such as sensor noise or quantization. cation. We use bilateral filtering to remove those.

Thankyou