DIP Project Proposal

Team: *Fourier and Dual*

Mohsin Mustafa (**20161131**)
Shyamgopal Karthik (**20161221**)

Project Title: Shadow Removal in Documents and Images
Github Link: `https://github.com/sgk98/Shadow-Removal`

Project Goal

The goal of this project is to create a image shadow-removal module for three cases: automatic, user-assisted and document-specific shadow removal.

Problem Description

We consider the problem of removing the shadows in an image using various image processing techniques. This is a frequently-occurring problem in photography, usually when the light source is behind the camera.

We want to approach three variants of this problem:

1. **Automatic Shadow Removal**: The goal is to be able to detect and remove shadows from any given image automatically without any assistance.

2. **User Assisted Shadow Removal**: The user gives input through a mouse click of the position of the shadow in the image. Since we can localize the shadow through user assistance, the problem is simplified. We would therefore expect better performance from this method as compared to [1].

3. **Shadow Removal in Documents**: We remove shadows from a document which has only text on it. Since we have priors about the image being a document, we want to achieve automatic shadow removal on documents which should give better results as compared to [1] and [2], for documents specifically.

All the three variants mentioned above have their use cases — for example, although user-assisted shadow removal is expected to give better results for a general image compared to automatic shadow removal, the latter

method is better for performing shadow removal on a huge database of images since it is a painstaking task to provide annotations on every image regarding the positions of shadows.

The following is a more technical description of the approach for each shadow-removal variant:

1. **Automatic Shadow Removal**: The additional challenge in automatic shadow removal is that we do not have any prior information about the location of the shadow, or of the type of image. The challenge here is that the intensity values alone cannot tell us if a region is in shadow. The key insight here is that pairs of regions with same illumination can be joined together. A graph is constructed where the edges are pairs with regions with same illumination. A graph cut algorithm is applied along with some post processing to finally get the shadow mask. Once the shadow mask has been computed, we can use the shadow removal technique from the below section to reconstruct the shadow free image.

2. **User Assisted Shadow Removal**: The input received here is a mouse click on a surface which contains both shadowed and non-shadowed regions. We first grow a seed based on the user input (and a distance metric with pixels) to compute a shadow mask. After the mask has been computed, we use the illumination model to reconstruct the shadow free image. We will have to make use of a Laplacian Pyramidal model to ensure that the reconstructed image is not noisy and is free from artifacts.

3. **Shadow Removal in Documents**: The key idea here is from intrinsic image decomposition (*i.e* splitting the image into reflectance and shading) for this purpose. After binarizing the image, we have to estimate the shading image and recover the reflectance image. Since some hard shadows still remain, the reflectance image is passed back and this process is repeatedly done until the shadows are removed. After this, tone mapping is done to ensure that the original texture and illumination of the document is not lost.

Final input and output: The module in current expectation will be invoked in the following way, and the output will be saved in the current directory.

```
python ShadowRemoval.py <shadow-removal-variant> <op-flname>
<if-applicable-shadow-region>
```

If the project is completed earlier than expected, a simple GUI may be made (through a web-app) for better usage of the shadow removal module.

Expected Results

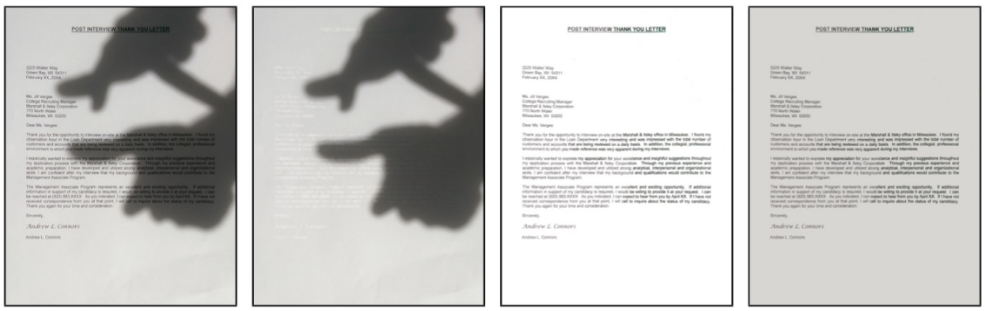

Figure 1: Shadow Removal in Documents

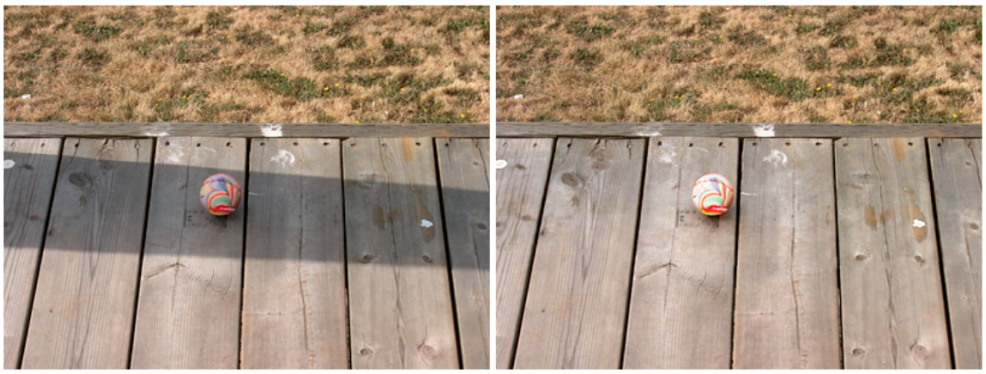

Figure 2: Shadow Removal in a General Image

Splitting of Tasks

Broadly, the topics taken up by each of us will be:

- Mohsin Mustafa
  - User-Assisted Shadow Removal
  - Shadow Detection for Automatic Shadow Removal
- Shyamgopal Karthik
  - Shadow Removal in Documents
  - Removing the detected shadow for Automatic Shadow Removal

The other tasks like integration of all the variants and a possible GUI for the module will be taken up such that there is equal work from both of us. Also, as and when necessary, the above listed tasks may require both of our contribution.

Milestones and Timeline

Sep 29	Submission of Proposal
Oct 1 - Oct 20	Work on and complete Document Shadow Removal and User Assisted Shadow Removal (can be worked on in parallel)
Oct 20 - Oct 30	Shadow Detection for Automatic Shadow Removal
Nov 1 - Nov 10	Automatic Shadow Removal completion
Nov 10 - Nov 15	Integration of the entire module
Nov 15 - ...	Compiling results, preparing report and presentation.

Effort will be put in so that work is done well ahead of the above schedule, so that a GUI can be possibly implemented.

References

These are the papers we will be following for this project:

Automatic Shadow Removal: R. Guo, Q. Dai, D. Hoiem: *Paired Regions for Shadow Detection and Removal*

User Assisted Shadow Removal: Y. Shor, D. Lischinski: *The Shadow Meets the Mask: Pyramid-Based Shadow Removal*

Document Shadow Removal: V. Shah, V. Gandhi: *An Iterative Approach for Shadow Removal in Document Images* • S. Bako, S. Darabi, E. Schechtman *et al.*: *Removing Shadows from Images of Documents*