DIP Project Report

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

Contents

Introduction

Problem Statement

The problem focuses on localizing the shadows in the image (shadow detection) as well as removing the shadows from the image (shadow removal). While removing the shadow, the goal is to maintain the texture of the shadowed regions, while only changing the illumination of the regions. For the scope of project, we consider two types of images:

1. **Natural Scene Images**: Here we assume a general image taken and try to remove shadows from it. Our aim is to compare two methods: point-provided interactive shadow detection, and automatic shadow detection.

2. **Document Images**: Here, we assume that the image is a printed document without figures or images.

Motivation

Shadow Removal can play a important role in Computer Vision. Many tasks in Computer Vision such as background subtraction and object detection can see improved performance if we perform shadow removal on the image as a pre-processing step.

Shadow Detection and Removal algorithms can also be an important component of image editing tools. A brief introduction to such a tool can be seen here.

Shadow Removal in Document Images has a very obvious application, which is to boost the performance of Optical Character Recognition (OCR). While the goal of Shadow Removal is to maintain the aesthetic structure of the image while removing the shadows, this can be used as a pre-processing step before binarizing the image. One relatable example is for removing a distracting shadow due to the light source being present behind the photographer.

The problem of shadow detection and shadow removal is also very closely related to the problem of Intrinsic Image Decomposition. In Intrinsic Image decomposition, we decompose the image into reflectance and shading such that $I(x, y) = R(x, y) \cdot I(x, y)$ where R is the reflectance, and S is the shading. The idea is that, since the illumination information is captured in the shading map, we would be able to use this information to do shadow removal. Intrinsic Image Decomposition is a more general problem while shadow removal is far more specific and can even be considered as an application of intrinsic image decomposition.

Overview

For the problem of shadow detection on natural images, we explore both interactive and automatic shadow detection. Interactive shadow detection takes a point from the user which is in a shadow, and uses this information to compute a shadow mask.
For Automatic Shadow Detection, we seek to generate the binary shadow mask purely by looking at the image alone. Once we have the shadow mask(using either approach, we relight the image so as to remove the shadow.

For Document Shadow Removal, given a printed document image, we obtain the shadow free image while maintaining the texture of the original image. We pose this problem as an Intrinsic Image Decomposition Problem

Document Shadow Removal

Approaches Considered

We apply the idea of using Intrinsic Image Decomposition to perform shadow removal on document images [10]. The key idea behind this approach is that in a document, the Reflectance can take only 2 values: 0 or 1. This comes from the fundamental structure of a printed document; there is either a black text or the white background. The algorithm is explained in the next subsection.

Work Done

The algorithms has 3 stages:

Binarization

We binarize the image I into 2 sets, foreground F and background B. This can be done using adaptive thresholding. However, we ensure that the threshold high so that there might be shadows classified as foreground, but no genuine text is classified as background. At the end of this step, we have a binary image M where

$$M(x,y) = 0, \forall (x,y) \in F M(x,y) = 1, \forall (x,y) \in B$$

.

Reflectance and Shading Estimation

To compute the Shading map, we use an approximation.
If $(x,y) \in B$, then we set $S(x,y) = I(x,y)$
This is because if it is in the background, the reflectance must be 1, and hence we get the shading value.
If it belongs to the foreground, we compute the shading map by interpolating the shading values around the the point. Formally,

$$S(x,y) = \begin{cases} I(x,y) & (x,y) \in B \\ \dfrac{\sum\limits_{i=x-dx}^{x+dx} \sum\limits_{j=y-dy}^{y+dy} I(i,j) \cdot M(i,j)}{\sum\limits_{i=x-dx}^{x+dx} \sum\limits_{j=y-dy}^{y+dy} M(i,j)} & (x,y) \in F \end{cases}$$

Once, the shading map has been computed, we compute the reflectance image using the following equation

$$R(x,y) = \frac{I(x,y)}{S(x,y)}$$

The reason we perform all these steps is that the reflectance image is free of shadows. However, in the case of strong shadows, some shadows still remain in the image. Hence, we repeat the above 2 steps repeatedly passing back the reflectance image as the input. At the end of this, we obtain a reflectance image R that is free of image, a shadow map S and a binarized image M.

Tone Mapping

The background pixels in the reflectance image R have values close to 1. However, we still want to recover the original texture of the image. We threshold the shading map S using Otsu's method to obtain a shadow mask SM. We calculate the global mean gm as follows:

$$gm = \frac{\sum_{i=1}^{r} \sum_{j=1}^{c} I(i,j) \cdot M(i,j) \cdot SM(i,j)}{\sum_{i=1}^{r} \sum_{j=1}^{c} M(i,j) \cdot SM(i,j)}$$

We compute the final output image as follows:

$$\hat{I} = I \cdot gm$$

Results

We show the outline of Document shadow Removal as well as sample results in Figs1-4. Our results while not state-of-the-art are still very competitive. However, in a couple of images here, we can see that we are losing a bit of the text as well. This happens when the text is very small and dim. This is the only issue with this approach.

Discussion

We have implemented a method that removes the uneven illumination and shadows in documents here. The results are very satisfactory and are possibly even better than a few contemporary papers in this field barring[10]. The only issue behind the method is the over reliance and the lack of robustness with respect to binarizing the image. The limitations of this work is the fact that it does not work for documents with figures/images etc. A future direction would be to generalize this approach to documents with images within them.

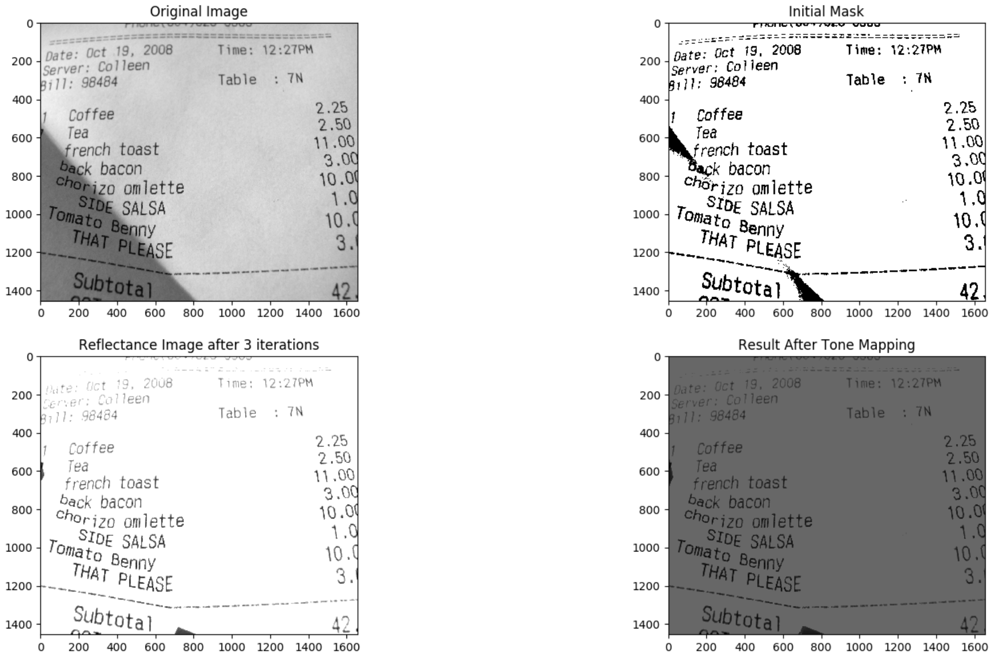

Figure 1: Overview of Document Shadow Removal

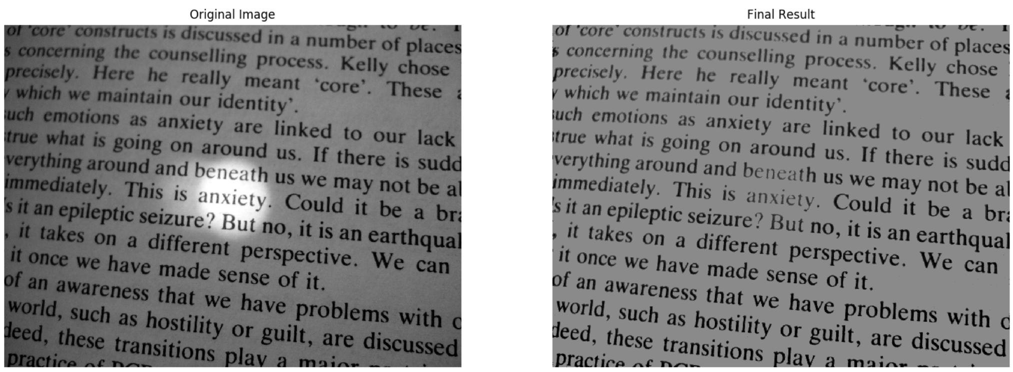

Figure 2: Sample of Results

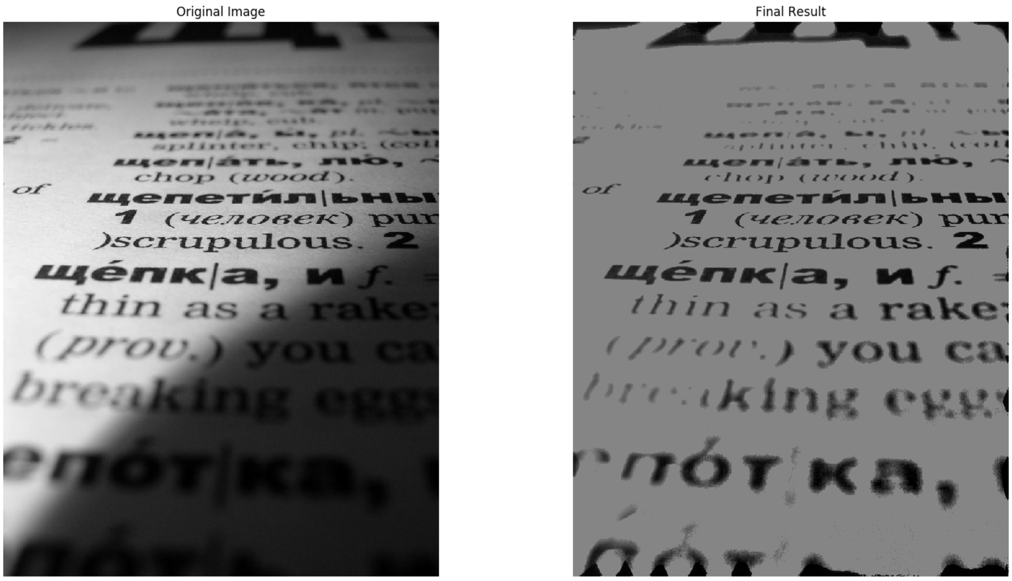

Figure 3: Sample of Results

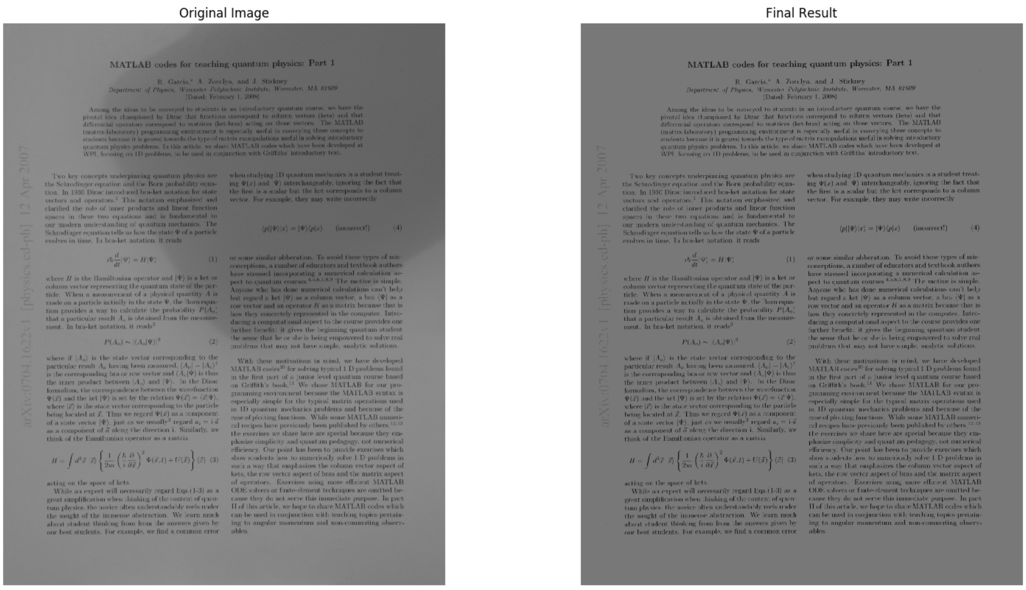

Figure 4: Sample of Results

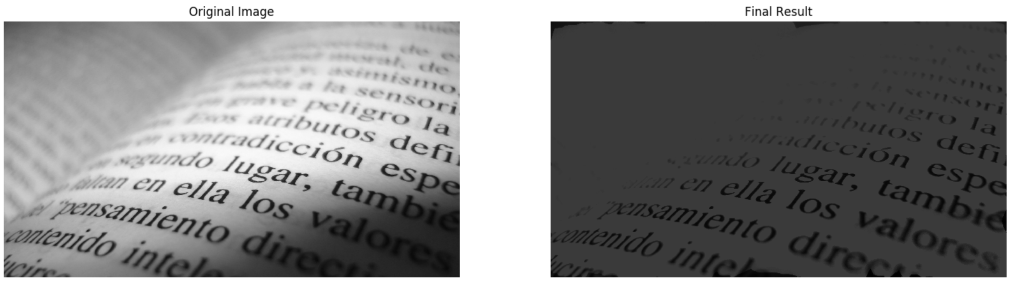

Figure 5: Failure Case where some text is lost

Natural Image Shadow Detection

For Natural Image Shadow Removal, we follow a two step procedure: Shadow Detection followed by Shadow Removal. This is a principled and modular approach to the problem since we can evaluate and benchmark algorithms for each step independently of each other. This section contains the shadow detection part of this procedure. This section is further divided into two: Interactive Shadow Detection and Automatic Shadow Detection.

Shadow Detection: Given a shadowed image I, the problem of shadow detection is to generate a Shadow Mask M such that

$$M(x,y) = \begin{cases} 1 & I(x,y) \in Shadow \\ 0 & I(x,y) \notin Shadow \end{cases}$$

Interactive Shadow Detection: The only input we take from the user is a seed point of a surface that contains both shadowed and non-shadowed regions.

Automatic Shadow Detection: The only input here is the original image.

Interactive Shadow Detection

Approaches Considered

Work Performed

The results of each sub-process in this interactive shadow detection algorithm are shown on two different example images. Then the complete pipeline is shown on one of those images. The pipeline starts off with the input image **Im**, and the provided point $\mathbf{P} = (\mathbf{P}_x, \mathbf{P}_y)$. **Im** is downsampled before the process begins. P must lie on a surface which has regions both in and out of the shadow.

Figure 6: The two example images used.

Shadow Seed

Figure 7: Generated Shadow Seeds.

Our first aim is to build up a small region around $\mathbf{P}$, called $\mathbf{S_s}$ or *Shadow Seed*, that satisfies the following properties:

- $\mathbf{S_s}$ is connected,
- $\mathbf{S_s}$ belongs to the same surface as $\mathbf{P}$,
- $\mathbf{S_s}$ is completely shadowed.

To satisfy these, we build $\mathbf{S_s}$ using breadth first search from $\mathbf{P}$ on a bilaterally filtered version of the image — to preserve edges while removing tiny noise-related fluctuations. A new pixel is added to $\mathbf{S_s}$ if its RGB value is only a tiny threshold (4 using Euclid distance) away from its neighbouring pixel already present in $\mathbf{S_s}$. The $\mathbf{S_s}$ mask is updated by morphologically closing it.[1]

Apart from providing a shadowed surface to start off with, $\mathbf{S_s}$ also has the task of providing a suitable estimate for the median colour of the surface $\mathbf{P}$ lies on, which is used in the next step of the pipeline.

Shadowed Region Surface

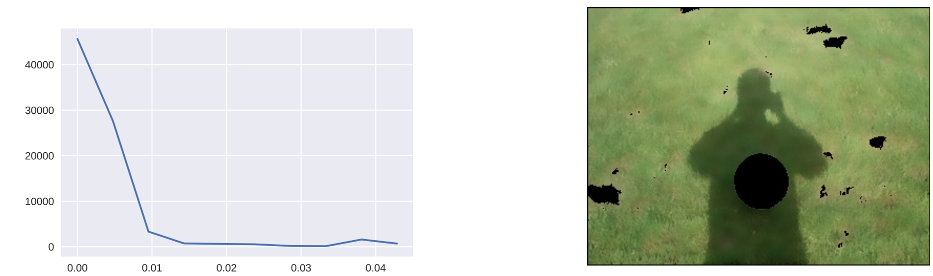

Figure 8: The first image with its surface region and histogram.

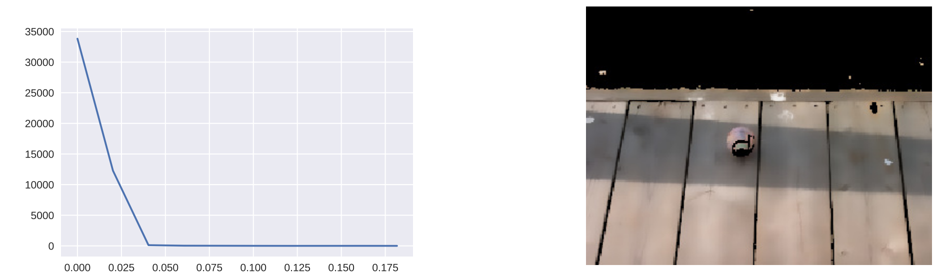

Figure 9: The second image with its surface region and histogram.

The next step is to estimate the surface of the region that $\mathbf{P}$ lies on. This is done by calculating an illumination-invariant distance of each pixel from the median colour of $\mathbf{P}$, and this distance between two colours is the following (as given in [5]):

$$|1 - \cos\theta| \text{ (where } \theta \text{ is the angle between the RGB vectors)}$$

The histogram of performing this operation on various images showed a sharp peak followed by low values[9]. A threshold that appears after the peak

[1]This is not mentioned in the consulted paper, but it gave better results.

is taken, and this threshold more or less divides the image into two: areas that belong to the surface, and areas that do not. Though the method to find the right threshold is not provided, we perform a rough estimation (which can be improved in some cases by manual work). This region belonging to the surface we now call $\mathbf{S_r}$.

$\mathbf{M_s}$ and $\mathbf{M_l}$

Figure 10: The first image with its generated $\mathbf{M_s}$. Left is luma and right is chroma.

Figure 11: The second image with its generated $\mathbf{M_s}$. Left is luma and right is chroma.

The next step is to divide $\mathbf{S_r}$ into two regions $\mathbf{M_s} and \mathbf{M_l}$ such that:

- $\mathbf{M_s}$ is shadowed,
- $\mathbf{M_l}$ is unshadowed,
- $\mathbf{M_s} \cup \mathbf{M_l} \subset \mathbf{S_r}$ and $\mathbf{M_s} \cup \mathbf{M_l} \lessapprox \mathbf{S_r}$.

Our initial underestimate for $\mathbf{M_s}$ is $\mathbf{S_s}$. Then the *closest* points in $\mathbf{M_l}$ are slowly added to $\mathbf{M_s}$ based on two similarity measures: one using the luma channel, and the other using the chroma channels. The points are added until standard deviation of $\mathbf{M_s}$ crosses that of $\mathbf{M_l}$.

The two $\mathbf{M_s}$ estimates (luma and chroma) are then conjoined into one by intersection.[2]

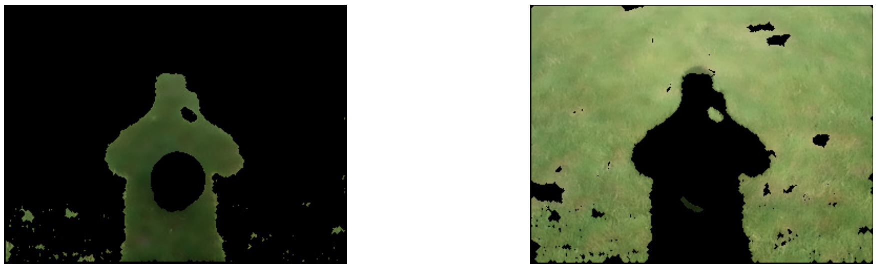

Figure 12: Final $\mathbf{M_s}$ and $\mathbf{M_l}$ for first image.

Figure 13: Final $\mathbf{M_s}$ and $\mathbf{M_l}$ for second image.

Matting and $\mathbf{M_{shadow}}$

We now use the computed subregions to convert the entire image into three categories in the following order:

- $\mathbf{M_s}$ and regions directly surrounding it, but not in $\mathbf{M_l}$, are categorised **definitely shadowed**.
- $\mathbf{M_l}$ is dilated and then classified **definitely unshadowed**.

[2]The paper suggested taking the one with fewer connected components, but that performed worse on our experiments.

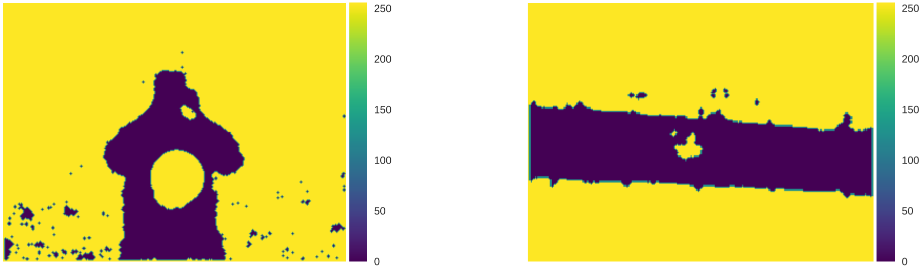

Figure 14: Pre-matting trimaps for both images.

- The rest are classified as **undecided**.

A trimap is built using these three categories, and this trimap and the original image is used for performing alpha-matting.[3]

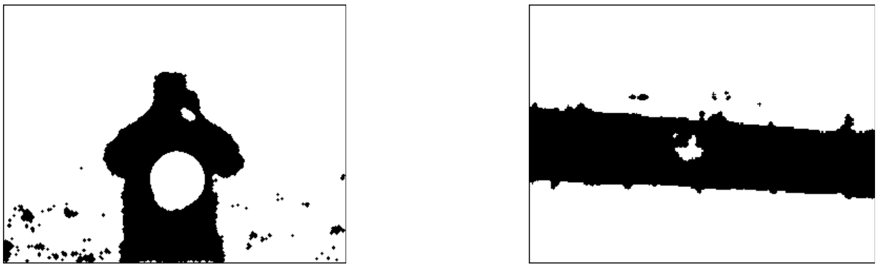

Figure 15: Final shadow masks after alpha-matting for both images.

Automatic Shadow Detection

The goal of Automatic Shadow mask is for a given Image I return a binary mask M such that

$$M(x,y) = 1, \forall(x,y) \in SM(x,y) = 1, \forall(x,y) \in B$$

. where B and S are the bright and shadowed areas respectively.

[3]Mishima's method [8] was used instead of the mentioned method.

Approaches Considered

We did consider using an SVM or some other classifier to classify regions of the image into shadowed and unshadowed regions respectively. However, we settled on a simple Image Processing Pipelines using intensity,colour and texture features.

Work Performed

As before, the results at various steps in the pipeline are shown for two example images. The pipeline starts off with a single image **Im** being provided.

Mean Shift Segmentation

The image is divided into segments using mean-shift segmentation as outlined in [2]. This method lies on matching each of these segments to either shadowed or not, based on their color-space features and their relative features with adjacent segments.

Matching Segments

Each segment is matched to another segment in the image based on three features: distance between centroids, gradient difference and texture difference. The gradient vector for an image is found by binning the gradient (Sobel) magnitudes of a segment into twenty parts, and finding the histogram. Similarly the textural histograms were founded by using a textons filter bank as provided by [7]. The difference between segments i and j is given by

$$\mathbf{difference}_{i,j} = ||\mathbf{centroid}_i - \mathbf{centroid}_j||^2 + |\mathbf{grad}_i - \mathbf{grad}_j| + |\mathbf{text}_i - \mathbf{text}_j|$$

Segment i will be matched to the segment j such that $\mathbf{difference}_{i,j}$ will be minimized.

Initial Shadowed Regions

We need to initialize the formulated segments with a set of shadowed segments, which has a high probability of being shadowed. Based on the ideas presented in [11], the segments with 0.6% the mean luma of the entire image should lie in the image.

Iterative Gaussian Algorithm

Now KMeans is performed on the mean normalized H/V to get clusters, and the assumption is that these two cluster centers will be the centers of the shadowed region and the unshadowed region. Then an iterative algorithm is performed — in each iteration, the segment with the highest probability of being shadowed, but not yet labeled a shadow, is taken. This segment is labeled a shadow, and the nearest segment to it is labeled

- **shadow** if their Gaussian probabilities are close,
- **non-shadow**, otherwise.

Finally, a last check is made on whether an unshadowed segment is similar to it's nearest shadowed neighbour based on value and luma features. If they indeed are similar, the unshadowed segment is made a shadow.

Results

We show the detected shadow masks. This stage of the algorithm performs well as long as the image segmentation algorithm performs well.

Figure 16: Shadow Mask for Automatic Shadow Detection

Discussion

We have come up with an extremely simple and elegant method for detecting and localizing shadows in an image. Our algorithm performs very well and computes the mask effectively in most cases. The only issue in our method is the over reliance on the segmentation. We have seen that the choice of parameters matter a lot for the performance and the optimal parameters differ from image to image. This is the only weakness in our algorithm. This approach works well on images with hard shadows. That is the transition from the umbra of the shadow to the unshadowed region is abrupt. In cases of soft shadows(where the shadow has a penumbra), we would need to modify our algorithm. This can be addressed in future works.

Figure 17: Hard Example for Automatic Shadow Detection

Natural Image Shadow Removal

Approaches Considered

The goal of this stage is given a shadow mask M, we need to compute the shadow-free image. We primarily approached the problem of relighting shadowed regions using the idea of paired regions [6]. The key idea is that we can do some form of illumination transfer for a shadowed region if we can pair it with an unshadowed region[11].

Work Done

Segmentation

The first task is to segment both the shadowed and unshadowed region.
We experimented with both mean shift clustering as well as graph based segmentation [3] [4].
We notice that while both of them do not give wildly different results, the parameters of the segmentation have to be chosen appropriately in either case to get a good segmentation of the unshadowed region.

Region Matching

For every region, we compute a set 3 values: the texture histograms computed either through a Gabor Filter Bank/Textons, the gradient histogram as well as the centroid of the region. The distance function between 2 regions is the Sum of Squared differences of these 3 values. For every shadowed region S, we assign a non-shadowed region N such that $dist(S, N)$ is minimum.

Relighting

Now we derive the shadow formation model and show the equation used to relight the image. This relighting is done in the LAB colour space since it is perceptually uniform so applying a linear increase on the values does not affect the image drastically. We know that an image I can be represented as $I(x) = R(x) \cdot L(x)$ where R is reflectance and L is the illumination. We can rewrite this as

$$I^{lit}(x) = R(x) \cdot (L^a + L^d)$$

where L^a is the ambient light and L^d is the direct light. In the case of the shadowed region, this would be:

$$I^{shadow}(x) = \eta \cdot R(x) \cdot L^a$$

Since the direct light is blocked and the ambient light would be attenuated(η is the attenuation factor). We now formulate an affine transformation to recover I^{lit}

$$I^{lit}(x) = R(x) \cdot L^a + \frac{1}{\eta} \cdot I^{shadow}$$

This can be re-formulated as:

$$I^{lit}(x) = A^{avg} + \gamma(x) \cdot I^{shadow}$$

where A^{avg} is $\mu(B) - \gamma(x) \cdot \mu(S)$ where B and S are unshadowed and shadowed regions respectively. $\gamma(x)$ is $\frac{\sigma(B)}{\sigma(S)}$ where μ and σ have their usual meanings of mean and standard deviation respectively. To further improve our results we compute A adaptively for each pixel, by

$$A = A^{avg} \cdot \frac{I(x)}{I^a vg}$$

where $I(x)$ is the value of the pixel and I^{avg} is $\mu(S)$ Using the above formulae, we recover the shadow-free image.

Boundary Processing

Once the relit image has been computed, we see that it has boundary artifacts. This is because an intensity transformation has been applied on one side of the boundary, but not on the other. To fix this artifact, we compute a boundary mask by performing morphological operations on the shadow mask M. Using this shadow mask, we inpaint the boundary regions using Adobe's Content Aware Fill(which is based on PatchMatch [1]). The resulting final image is smooth and free of boundary artifacts.

Results

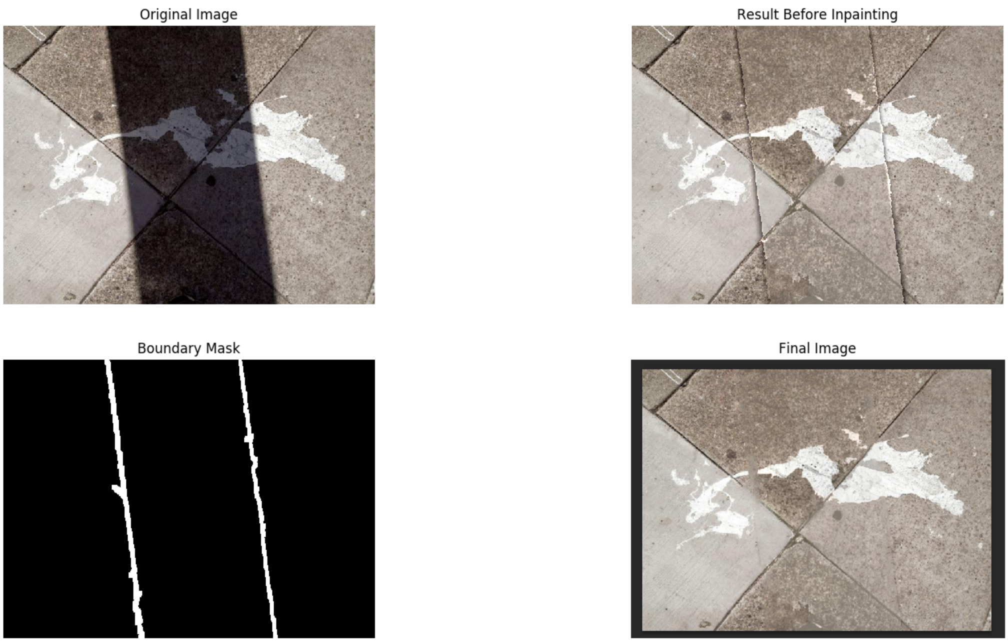

Figure 18: Overview of Shadow Removal Pipeline

We show the overview of our removal pipeline. The key behind the success of this method is the subregion matching. This does not extend very well to complex scenes with a lot of clutter. And we also illustrate a failure case where the region matching when the subregion matching does not work well.

Figure 19: Sample Results for Shadow Removal

Figure 20: Failure Case for Shadow Removal Due to bad region matching(Boundary Processing not done here)

Discussion

The above method presented for shadow removal works well on reasonably simple scenes. However, the 2 main limitations of this method is that they do not work so well on soft shadows and they don't generalize well to complex scenes. There is a lot of scope to pursue this problem further and it still remains an active area of research in 2018.

Work Split

Task	Shyam's Contribution	Mohsin's Contribution
Document Shadow Removal	90%	10%
Interactive Shadow Detection	10%	90%
Automatic Shadow Detection	30%	70%
Automatic Shadow Removal	70%	30%
Report	50%	50%

Table 1: Splitting of Work

Off the Shelf Implementations Used

We made liberal use of Skimage functions. The other important off the shelf tools used were:

- **Graph Based Segmentation [4]**
- **Mean Shift Clustering [2]**
- **Inpainting by Adobe Photoshop [1]**

References

[1] Connelly Barnes et al. "PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing". In: *ACM Trans. Graph.* 28.3 (July 2009), 24:1–24:11. ISSN: 0730-0301. DOI: 10.1145/1531326.1531330. URL: http://doi.acm.org/10.1145/1531326.1531330.

[2] Dorin Comaniciu and Peter Meer. "Mean shift: A robust approach toward feature space analysis". In: *IEEE Transactions on pattern analysis and machine intelligence* 24.5 (2002), pp. 603–619.

[3] Dorin Comaniciu and Peter Meer. "Mean Shift: A Robust Approach Toward Feature Space Analysis". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 24.5 (May 2002), pp. 603–619. ISSN: 0162-8828. DOI: 10.1109/34.1000236. URL: http://dx.doi.org/10.1109/34.1000236.

[4] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. “Efficient Graph-Based Image Segmentation”. In: *Int. J. Comput. Vision* 59.2 (Sept. 2004), pp. 167–181. ISSN: 0920-5691. DOI: `10.1023/B:VISI.0000022288.19776.77`. URL: `https://doi.org/10.1023/B:VISI.0000022288.19776.77`.

[5] Graham D. Finlayson, Steven D. Hordley, and Mark S. Drew. “Removing Shadows from Images”. In: *Proceedings of the 7th European Conference on Computer Vision-Part IV*. ECCV ’02. London, UK, UK: Springer-Verlag, 2002, pp. 823–836. ISBN: 3-540-43748-7. URL: `http://dl.acm.org/citation.cfm?id=645318.649239`.

[6] Ruiqi Guo, Qieyun Dai, and D. Hoiem. “Paired Regions for Shadow Detection and Removal”. In: *IEEE Transactions on Pattern Analysis Machine Intelligence* 35.12 (2013), pp. 2956–2967. ISSN: 0162-8828. DOI: `10.1109/TPAMI.2012.214`. URL: `doi.ieeecomputersociety.org/10.1109/TPAMI.2012.214`.

[7] D. Martin et al. “A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics”. In: *Proc. 8th Int’l Conf. Computer Vision*. Vol. 2. 2001, pp. 416–423.

[8] Yasushi Mishima. *Soft edge chroma-key generation based upon hexoctahedral color space*. US Patent 5,355,174. 1994.

[9] Yael Shor and Dani Lischinski. “The Shadow Meets the Mask: Pyramid-Based Shadow Removal”. In: *Computer Graphics Forum* 27.2 (Apr. 2008), pp. 577–586.

[10] Shah Vatsal and Gandhi Vineet. “An Iterative approach for Shadow Removal in Document Images”. In: *ICASSP*. 2018.

[11] Chunxia Xiao et al. “Efficient Shadow Removal Using Subregion Matching Illumination Transfer”. In: *Comput. Graph. Forum* 32.7 (2013), pp. 421–430.