DIGITAL IMAGE PROCESSING PROJECT REPORT

INTRINSIC IMAGES IN THE WILD

Subramanyam(20161190) Sushman KVS(20161143)

Problem Statement

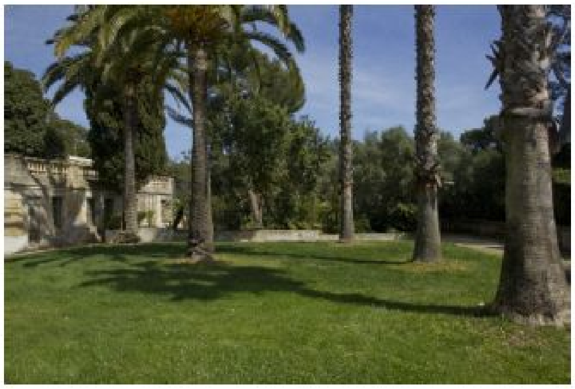
Intrinsic image decomposition is a long-standing inverse problem with many applications in graphics and vision. The goal of intrinsic images is to separate an image into two layers, a reflectance (albedo) image and a shading (irradiance) image, which multiply to form the original image. Reliable algorithms for separating illumination from reflectance in scenes would enable a range of applications, such as image-based resurfacing, texture transfer between images, relighting, material recognition, and other interior design tasks.

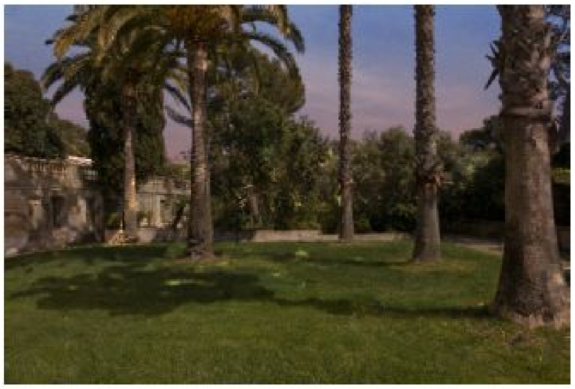
So we propose an algorithm which makes use of the fact that many surfaces in indoor scenes share the same material and reflectance, resulting in longrange sharing of reflectances across a scene (for example, a painted wall spanning an entire image). We build this algorithm on recent work in fully connected conditional random field (CRF) which was used for image detection and segmentation to enable such long-range connections in our algorithm.

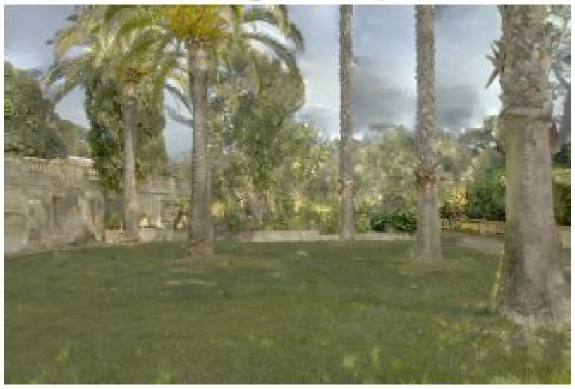
Motivation

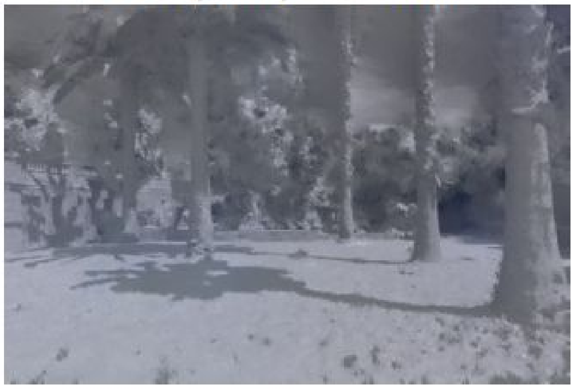
Intrinsic image decomposition is used in wide range of applications, some of them are explained as shown below:

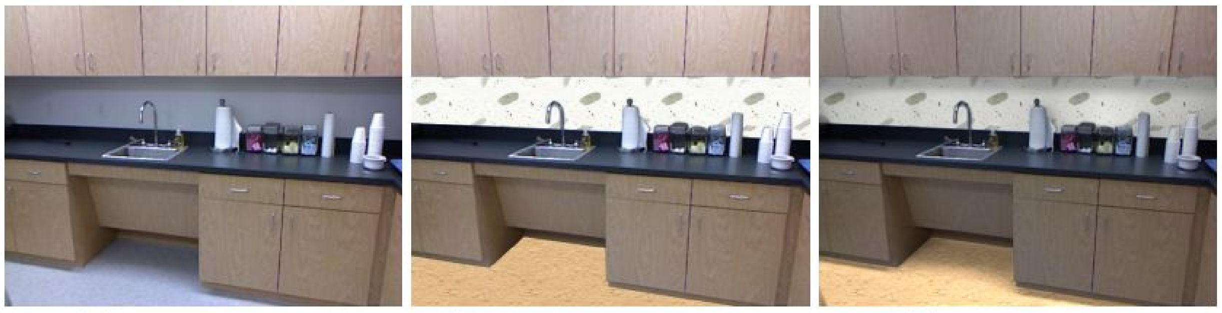
Intrinsic Images can be used in image relighting as shown in the figure. This is performed by gathering the intrinsic layers of the input image in different views. Then with the help of cast shadows relighting is performed on the image.


(a) Input image


(b) Relighting +30 minutes

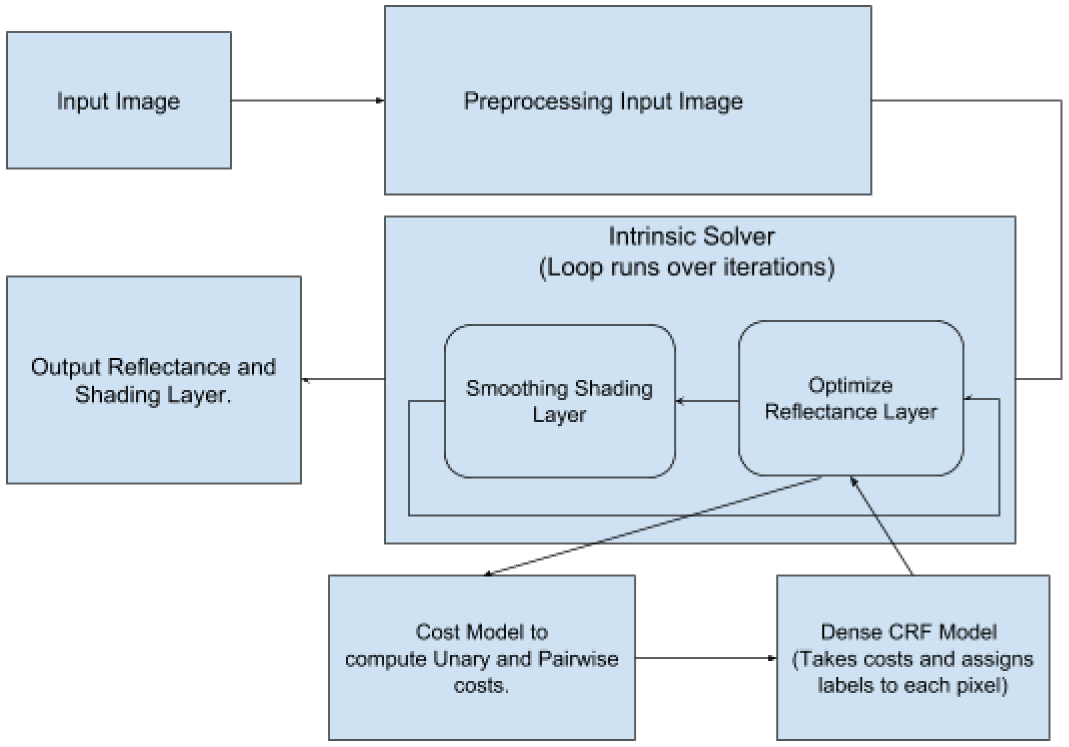

(c) Reflectance


(d) Shading


(a) Original image (b) Naive copy & paste (c) Our method

Intrinsic Image decomposition is also used in texture composition. Composing textures into images naturally is very tricky. because it requires careful consideration of spatially varying illumination. If illumination changes such as shadows are not properly handled, composition results may look too flat and artificial as shown in second image. Instead, we can first decompose an image into shading and reflectance layers, and compose new textures into the reflectance layer. Then, by recombining the shading and reflectance layers, we can obtain a more naturally-looking result as shown in the third image.

Algorithm Overview


Input Image
Preprocessing Input Image
Intrinsic Solver
(Loop runs over iterations)
Output Reflectance and Shading Layer.
Smoothing Shading Layer
Optimize Reflectance Layer
Cost Model to compute Unary and Pairwise costs.
Dense CRF Model (Takes costs and assigns labels to each pixel)

Pipeline

1. Image Preprocessing
2. Initializing hyper parameters and implementing cost functions.
3. Optimizing Reflectance Layer
4. Optimizing Shading Layer

Image Preprocessing

In this stage we will be storing our input image in a class model and declare necessary methods to be applied in that model. We will be declaring methods to map our input RGB image to various other spaces such as IRG(Intensity, Red chromaticity, Green chromaticity). We would compute a palette of reflectance colors, for this we do kmeans clustering in IRG space. We do clustering in IRG space because it is observed that chromaticity and intensity describe the pixels more properly than the RGB pixel colors. By applying kmeans algorithm we get the label centres. We use the intensities and chromaticities of these major k centres for the rest of the process. We would also writing methods to handle binary masks (neglecting some pixels of the input image).

Initializing hyperparameters and cost functions

Here comes the main assumptions. We have to tune some hyperparameters which will be used while optimizing the reflectance and shading layer. Such as the number of iterations for smoothening the shading layer, number of iterations required to run the denseCRF field. All the optimized parameters were already mentioned in the paper. We used them for building the model. Also cost functions are important while optimizing the shading and reflectance layer, there are mainly three cost functions used which would be explained with equations in each stage. Since the hyperparameters we used were mainly good for indoor scenes, our algorithm performs very good on indoor scenes than outdoor models.

Optimizing Reflectance Layer

In this stage we try to assign each pixel a label from the set of reflectances computer before using kmeans clustering. We use a fully connected conditional random field which takes the unary costs, pairwise costs and features associated with the pixels as input and returns the set of labels for each pixel which would minimize the total cost and increase the probability of reflectance layer defining the image. Unary costs and pairwise costs are computed by using the formulas as shown below:

$$E(x) = \underbrace{w_p E_p(x)}_{\text{pairwise } \psi_{ij}} + \underbrace{w_s E_s(x) + w_l E_l(x)}_{\text{unary } \psi_i}$$

We use intensity, chromaticity and pixel locations for computing these costs. We run the conditional random field over 10-15 iterations and get the optimized reflectance and the corresponding shading layer.

Optimizing Shading Layer

Smoothness optimization

Still we have to optimize our shading layer and smoothing is also important since we want the shading channel to very smoothly across smooth surfaces. Also for smoothing one more term for unary cost calculation is added in the DenseCRF model which would compute the cost for smoothing the shading layer. We solved optimizing the smoothing layer in an iterative fashion. The formula for unary cost in case of shading layer is given below:

$$E_s(x^{(t)}) = \sum_i \left(\log S_i^{(t)} - \log \tilde{S}_i^{(t-1)} \right)^2$$

$$\tilde{S}_i^{(t-1)} = \frac{1}{A_i} \sum_j S_j^{(t-1)} \exp\left(-\frac{1}{2} \left\| \frac{\mathbf{p}_i - \mathbf{p}_j}{\sigma_s^{(t)}} \right\|_2^2 \right)$$

Also for each iteration we will be gradually decreasing the gaussian kernel parameter using the below equation:

$$\sigma_s^{(t)} = \frac{\hat{\sigma}_s \, d}{t}$$

Here 'd' is the image diameter and 't' is the iteration number. Around 25 iterations were used for optimizing the shading layer.

Reducing shading discontinuities

To improve the shading intensity we have to minimize the discontinuities between adjacent regions across the image. Hence to do this we will be continuously optimize the palette of reflectances in order to improve the shading layer. Hence to say finally we even optimize the reflectance layer in each step iteratively and obtain the shading layer from the reflectance layer. All the formulas involved in this step are as shown:

$$\mathcal{R}^{(t)}(i) = r^{(t)}(i) \cdot \mathcal{R}^{(t-1)}(i)$$

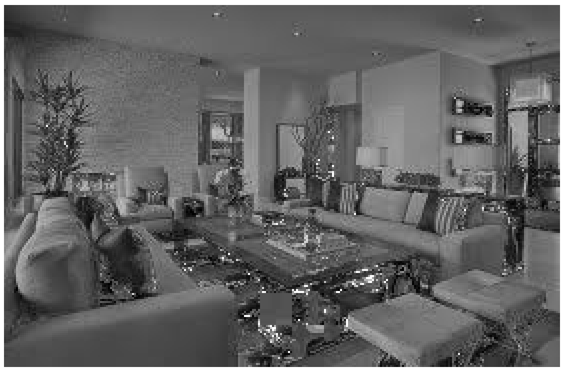
$$r^{(t)} = \arg\min_r \sum_{(i,j)\in B} |\log S_i - \log S_j| \tag{19}$$

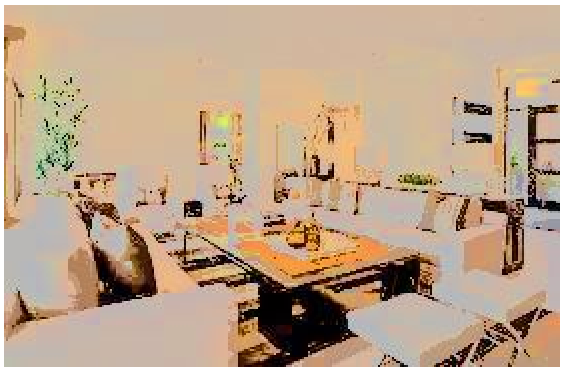
$$\log S_i = \log\left(\sum_c I_i^c\right) - \log\left(\sum_c \mathcal{R}^{(t-1),c}(x_i)\right) - \log r(x_i)$$

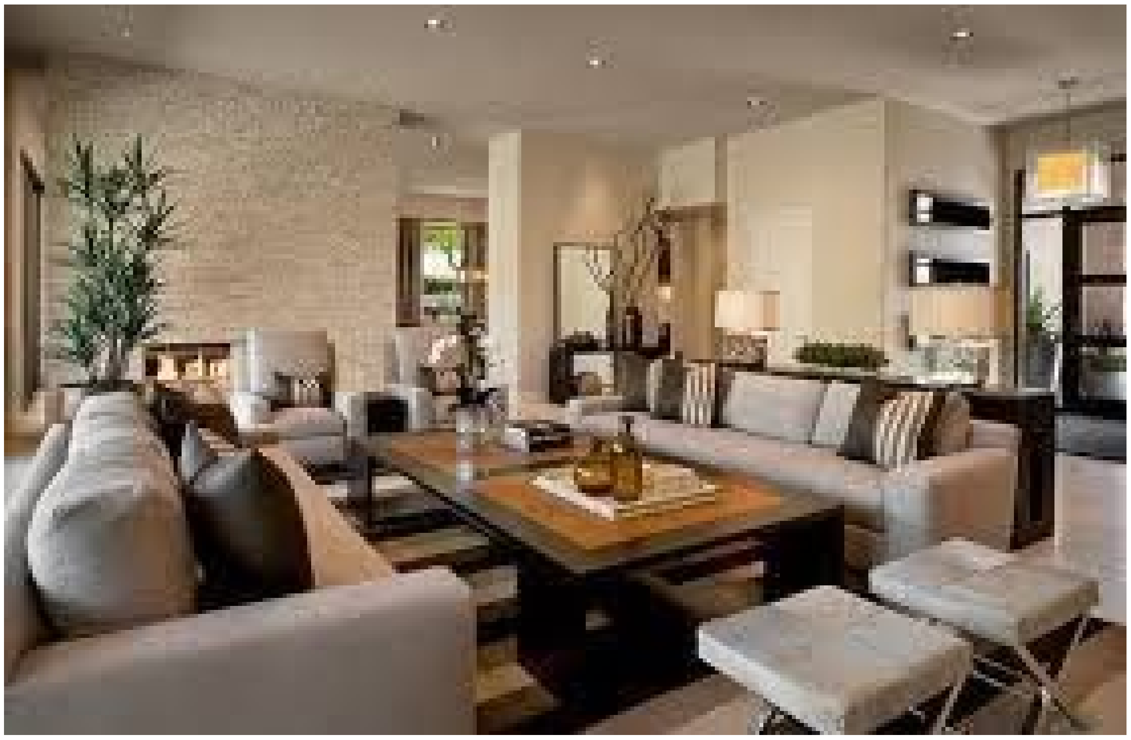
Scalar 'r' is used to regularize the reflectance layer, and these formulas are run over a number of iterations to get the final optimized shading and reflectance layer.

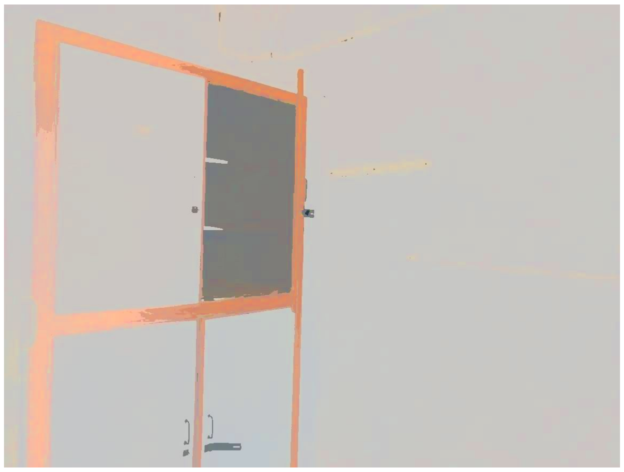
Results

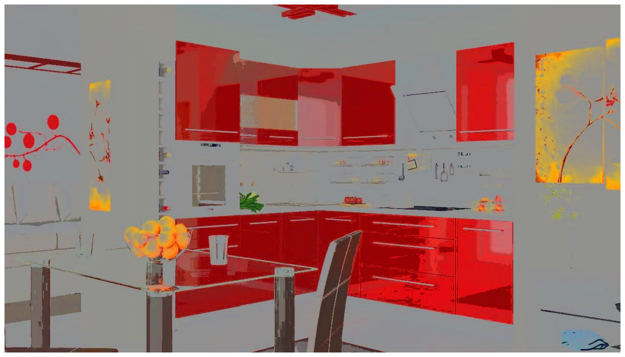
<u>MID EVALUATION RESULTS (Only reflectance layer was optimized)</u>





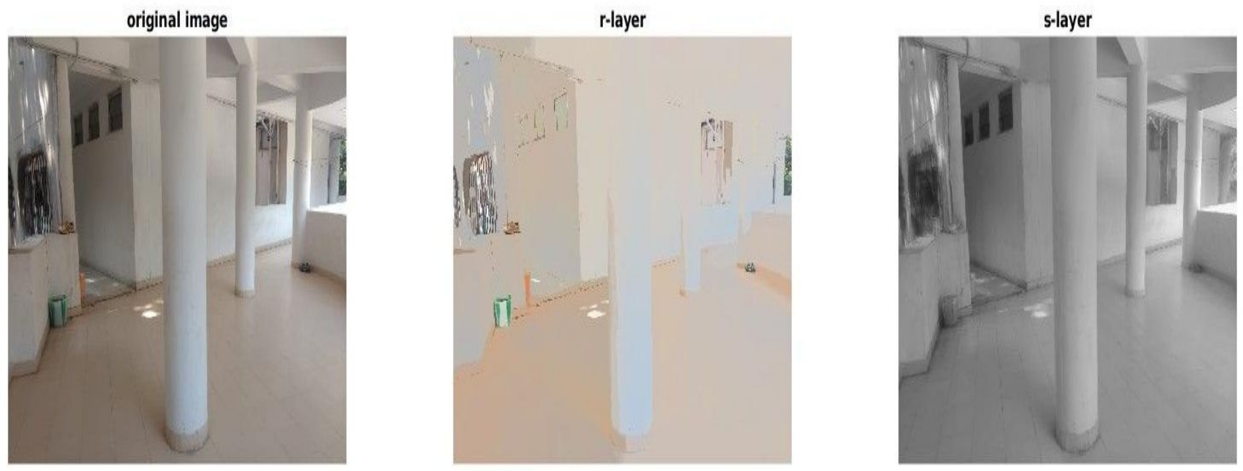


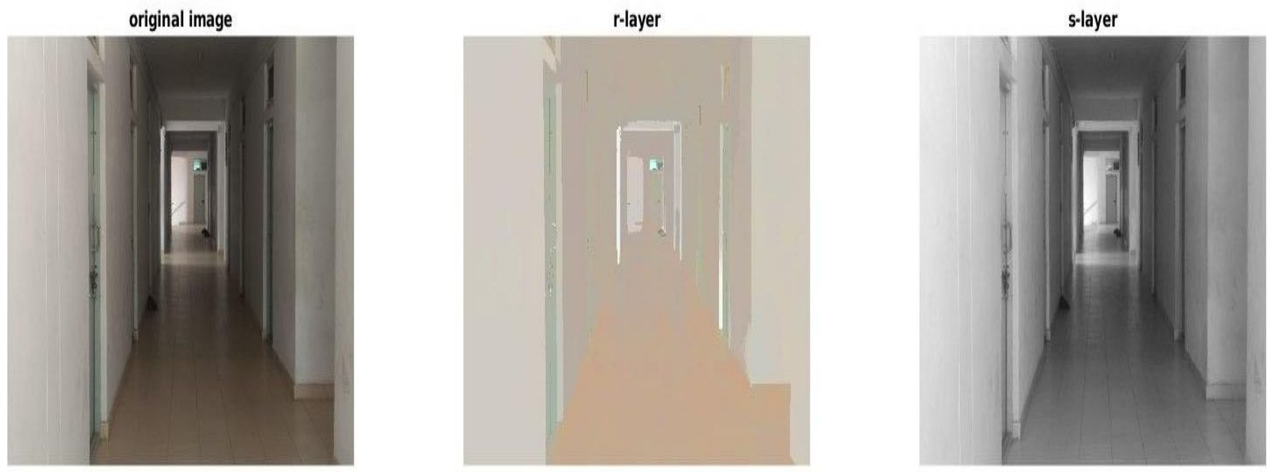


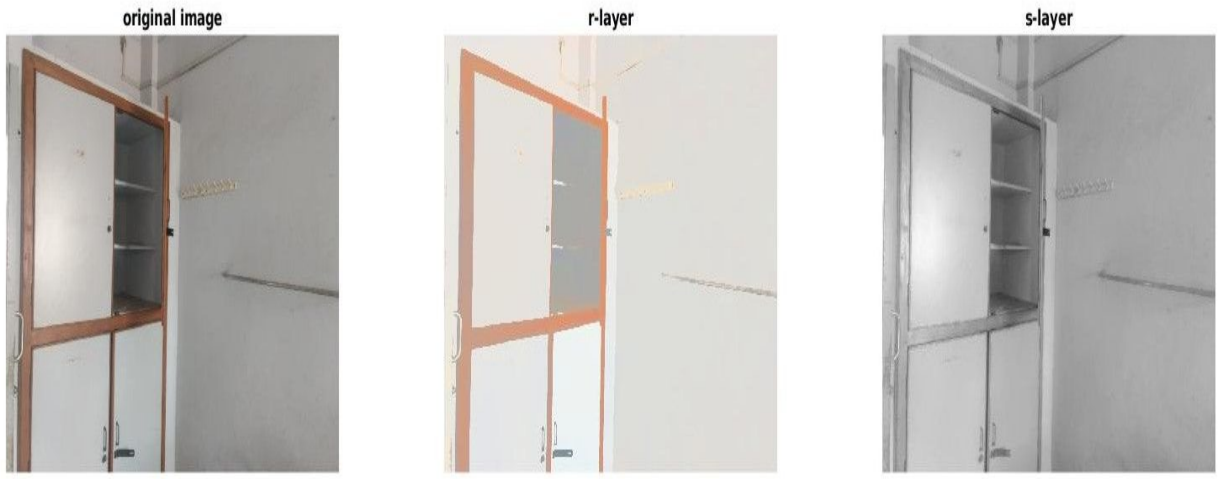


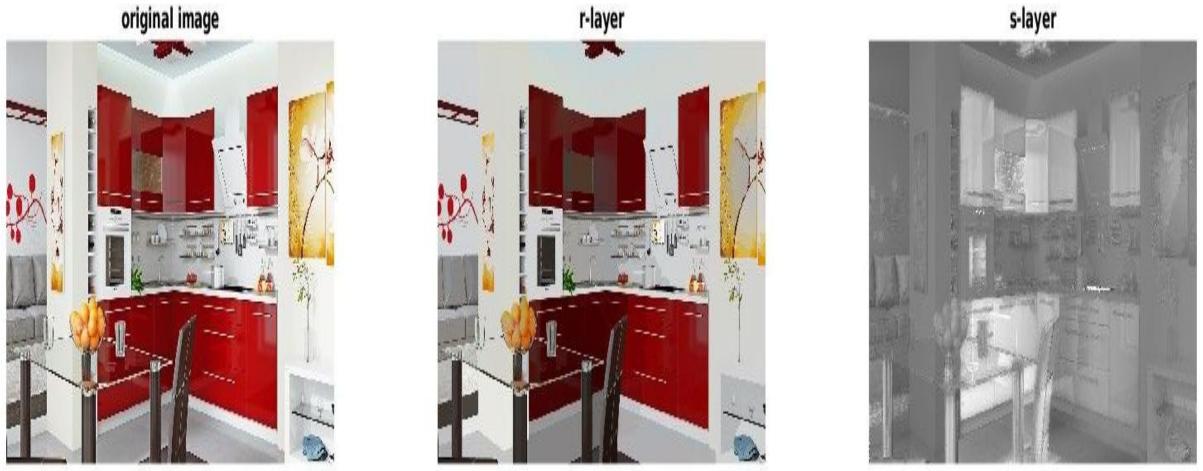
Final results after optimizing both reflectance and shading layers

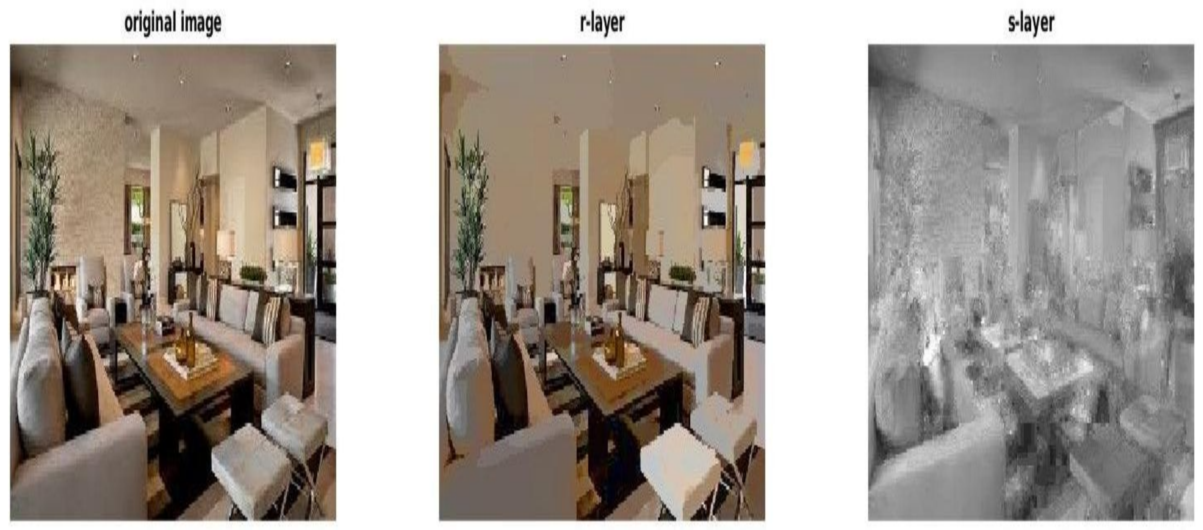
IMAGES FROM OBH

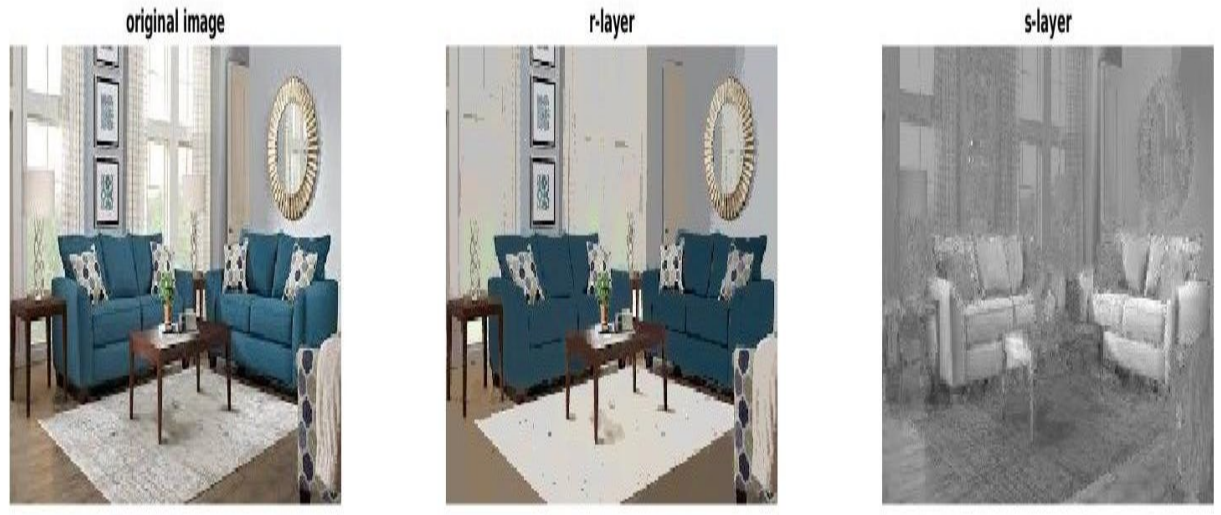


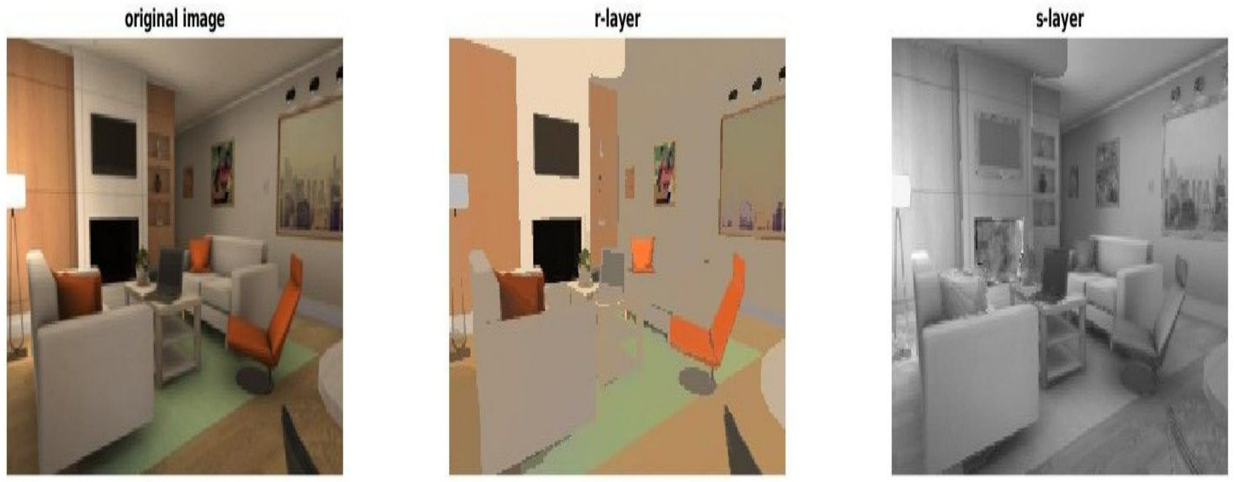



original image
r-layer
s-layer

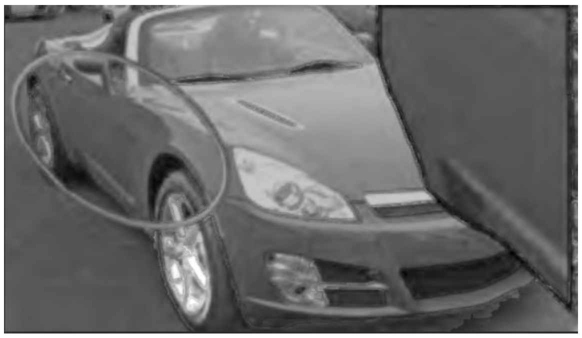

original image
r-layer
s-layer

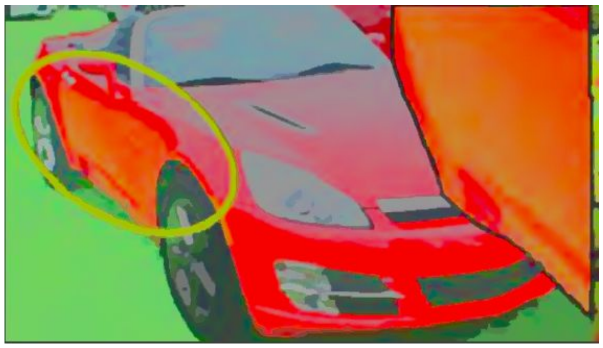

original image
r-layer
s-layer


original image
r-layer
s-layer



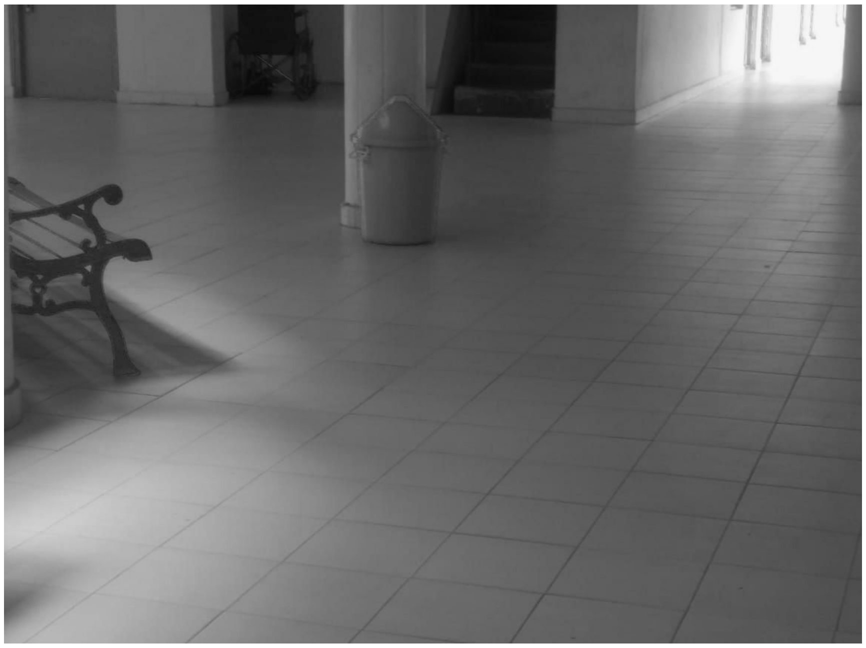
FAILURE CASES

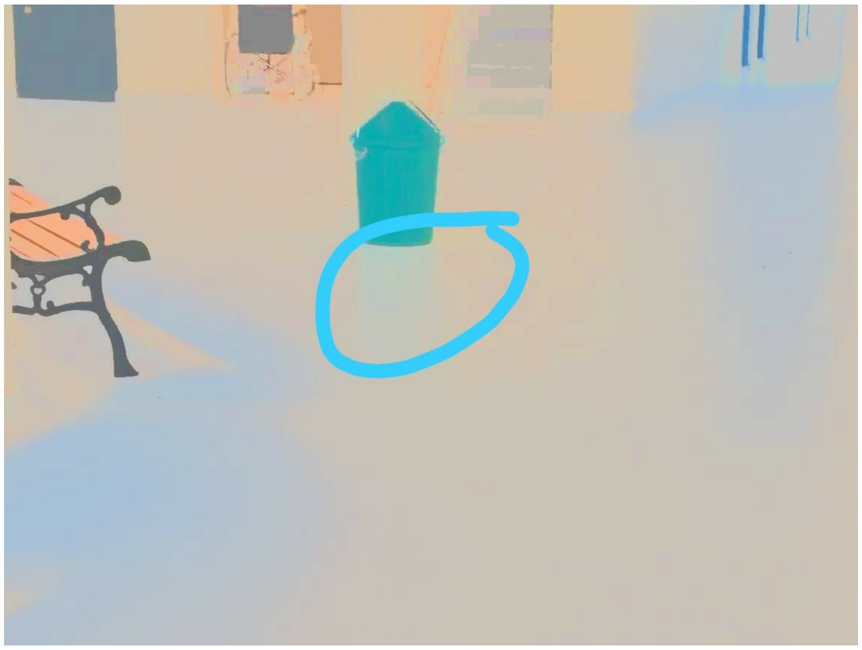




Colored Lighting

The intrinsic image algorithm we implemented was nicely filtering out mirror and transparent objects properly while obtaining the reflectance layer, however in case of real world objects have colored illumination especially when considering interreflections between colored surfaces. In the image above we can observe that there is green illumination over the car as rounded off, after decomposing the layers we can see that reflectance layer has green patches on the car.





Similar problem occurs in this image. There is a slight reflection of sea green color on the floor, (colored interreflection). After decomposing we observe that there is significant amount of green color on the floor in the reflectance layer.

FUTURE WORK

Presently the hyperparameters used in the algorithm were tuned to do a reasonable job on typical indoor photographs and very less on outdoor natural scenes. Therefore it does not represent the statistics of all images available online. Future work could be adding either more hyperparameters or adjusting the present priors to work well on outdoor photographs.

Github link: https://github.com/bruce-wayne99/Intrinsic-Images-in-the-Wild

TASKS ASSIGNMENT

Task	Team Member
Image preprocessing	Subramanyam MNS
Hyperparameters and tuning	Sushman KVS
Optimizing reflectance layer	Subramanyam MNS
Optimizing shading layer	Sushman KVS
Linking the pipeline	Both

REFERENCES

Research paper

DenseCRF model from github

Intrinsic Images in the Wild Website