
Computer Vision 1 - Assignment 1

Ruben-Laurentiu Grab

ruben.grab@student.uva.nl 11609923

Andrei Marius Sili

andrei.sili@student.uva.nl 11659416

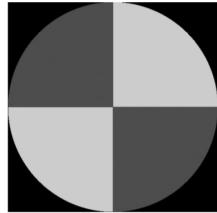
1 Introduction

In this assignment we experiment with reflection models, shading models, photometric stereo, colour spaces, intrinsic image decomposition and reconstruction, and colour constancy algorithms. We strive to provide a theoretical grounding on our results, or focus on observational and experimental data when that is not possible.

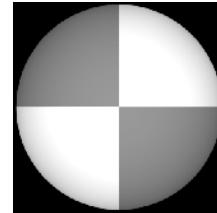
2 Photometric Stereo

2.1 Estimating Albedo and Surface Normal

Question 1.1 After completing the code for estimating albedo and surface normal map and running it on the *SphereGray5* data set, the actual results and the expected results differ to some extent. Specifically, we expected the albedo to consist of a circle divided in juxtaposed identical white and grey quarters. The algorithm generated a similar image, with the difference that towards the outside of the circle, one can observe a darker rim, as if this virtual object would be shaded. The figure below shows the expected and realised albedo representation.



(a) Expected albedo representation



(b) Realised albedo representation

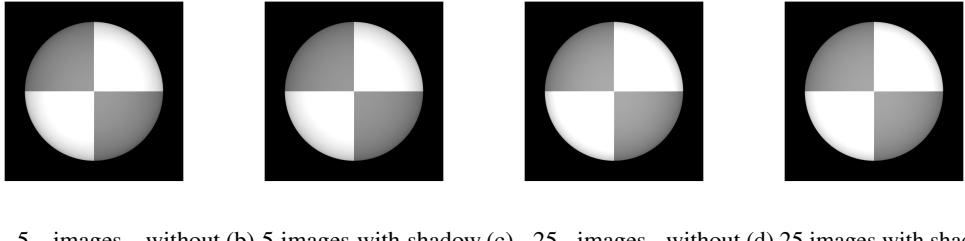
Figure 1: Expected and realised albedo recovery for 5 images

We believe one reason behind this distortion is the residual least squares error present after the recovery of the input radiance $\mathbf{g}(x, y)$. The error could arise as an effect of the fact that any point $P = (x, y)$ close to the edge of the sphere will have a darker tint for all instances in which the light source does not come from that particular side of the sphere because it is partly shaded. Although the vector \mathbf{V} accounts for this to some extent, assumptions of a local shading model and the point light source at infinity fail to capture the phenomena entirely. The system of equations from which the input radiance is extracted is in a sense not constrained enough. We explore this topic in more detail next.

Another reason could also be that the image corresponding to a light source centred on the sphere generates much brighter values, than even the brightest patches on the other images.

Question 1.2 In order to estimate the albedo and the surface normal, we need to solve the equation:

$$\mathbf{i}(x, y) = \mathcal{V}\mathbf{g}(x, y) \quad (1)$$



(a) 5 images without shadow trick (b) 5 images with shadow trick (c) 25 images without shadow trick (d) 25 images with shadow trick

Figure 2: Albedos with and without shadow trick

where the \mathcal{V} is a property of the illumination and of the camera and $\mathbf{g}(x, y) = \rho(x, y)\mathbf{N}(x, y)$ describes the surface. Of course, in order for this equation to be mathematically possible, the dimensions of the terms must be the same. Given that we work in a 3-Dimensional space, the size of $\mathbf{g}(x, y)$ and \mathcal{V} is 3. Thus, in the task of estimating albedo and the surface normals, for each pixel in the image we need to solve for 3 unknowns. In order to do this, we need at least 3 equations. Because of the images noise and for a better estimate of both albedo and surface normal, we can use more than 3 images and perform a least squares solution.

When running the algorithm using SphereGray25 we noticed a better estimate of the albedo. Comparing the results with those of SphereGray5, the darker rim is thinner when using more images, which is a better approximate of the sphere's albedo. We tested how the algorithm performs with different number of images, in an incremental way. We observed that the change in the quality of the albedo increases more rapidly in the beginning(i.e the difference between when running it with 21 and 25 of images is not as great as the difference from 13 to 17 images), and using around 19 images gives a decent result, comparing to the number of images used. One could use a strategy in which it checks to see when adding new images does not improve the estimation more than a certain threshold.

Question 1.3 In photo-metric stereo, shadows are problematic since they obscure the true pixel values of an object under proper lighting conditions. Consider a point P of an image i with a light source such that the pixel is shadowed, $I_P = 0$. Solving for the least squares solution of P , image i will yield an artificial constraint imposed on the input radiance $\mathbf{g}(x, y)$, since the true value of that point is completely obscured.

$$0 = \mathbf{V}^T \mathbf{g}(x, y) \quad (2)$$

For this reason, a ‘shadow trick’ is employed to neutralise this effect. Under the assumption of no ambient lighting, a diagonal matrix can be generated from the vector of point values of each image. We left-multiply the equation for pixel values by this matrix to give:

$$\mathcal{I}\mathbf{i}(x, y) = \mathcal{I}\mathcal{V}\mathbf{g}(x, y) \quad (3)$$

Consider now a set of 5 images, and an equation system for a point P such that the last image has value 0. This means that the last row of the matrix \mathcal{I} will be a $\mathbf{0}$ vector. Thus, the vector resulting on the RHS of equation 3 will also contain 0 as its last element, effectively zeroing the equation which was previously imposing an artificial constraint.

The experiments we run on this issue show mixed results. For a 5-image set, the shadow trick seems to worsen albedo recovery. We find this result counter-intuitive and don't have a reasonable explanation for it. For the 25-image set, there is little to no difference between using the shadow trick and not using it. This seems reasonable since we can assume that the set covers all variability in the object representation given various lighting conditions. This means that the shadowed points will have little overall impact on the least squares solution. Results are shown in figure 2.

2.2 Test of Integrability

Question 2 Using a threshold of 0.001, we can see that the edge of the sphere causes errors for both image sets (5 and 25). This corresponds to points along the sphere rim where the normals have a X or Y component much higher compared to the other components. This can be observed in the normal map, or more obviously in the X and Y decomposition 3. It seems that in order for the numerical

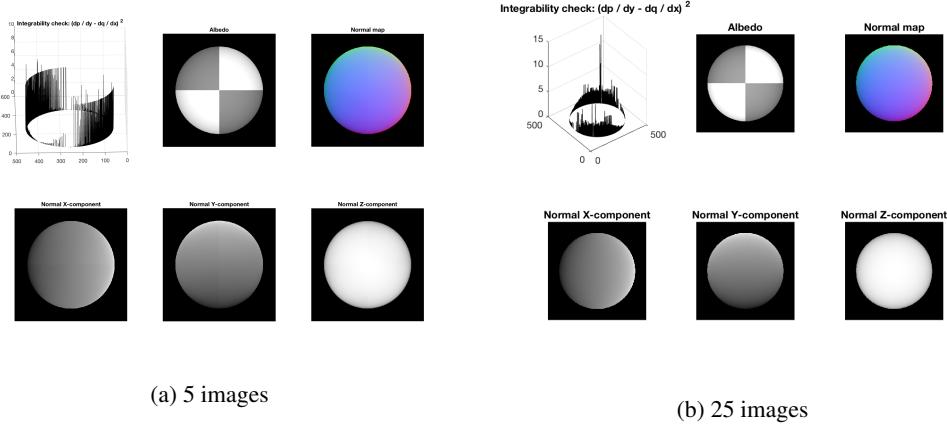


Figure 3: Integrability results

methods to provide reliable results, the X and Y components must vary together. With more images, the outliers decrease by about 1000, and the errors per outlier decrease drastically.

2.3 Shape by Integration

Question 3.1 When integrating by column, we observe that the resulting shape is distorted 'by row', whereas when integrating by row, the effect is similar, but distorted 'by column'. The reason this happens is that on the edges, the derivative are erroneous and the outliers are propagated by column or by row, depending on the strategy used. Figure 4 (b) shows that the derivative of $f(x, y)$ with respect to x contains values that are outliers. It also makes it easier to visualise how these values are distributed on the edge of the sphere. Because of these outliers, seen as spikes in figure 4 (b), which are summated when using column-major orders, the height map is distorted in such a way. Figure 4 (a) shows the effect that relative small errors in computing the first order derivative have on creating the shape of the object.

Question 3.2 When taking the average results of the two methods, we see a slightly different result, with features that are borrowed from both approaches: we see that the distortions appear to take place on both columns and rows, but in contrast to before, they appear not to be that significant. Figure 4 (a) and (d) helps to visualise the results. We also tried to run the algorithm on different data-sets. As expected, the results for when running it with Sphere25 outperformed the results for when using Sphere5. Figure 4 (c) and (d) shows us the manner in which adding more images yields better results: the propagated errors are much smaller, and the shape of the sphere seems smoother. In conclusion, adding more images gives us smaller errors, and in addition, using the average of the two methods reduces even more the extent of irregularities, but with the small cost of adding more of them.

2.4 Experiments with Different Objects

Question 4 Apart from the common errors that albedo is slightly shaded around the edges of the face, a new set of errors arise around the grooves representing the eyes, mouth, and ear interiors. These areas should have the same albedo as their neighbours, but because much less light reaches those spots, the albedo seems slightly shaded. This issue is presented in figure 5.8 of the 'Sources, Shadows, and Shading' material. The problem lies in the assumption of a constant ambient lighting, or no ambient lighting. Since a patch inside the groove 'sees' much less of the world than other parts of the object, it will experience substantially different ambient lighting compared to other patches.

We show our experiments in figure 5. With only 55 of the 121 images picked such that the light source covers the entire surface of the object similarly, the albedo reconstruction is quite similar to the reconstruction from the full set of images. However, when picking 55 images with lighting only from the left side, the right side of the monkey face shows many artefacts, especially around the grooves of the eyes and the edges of the face.

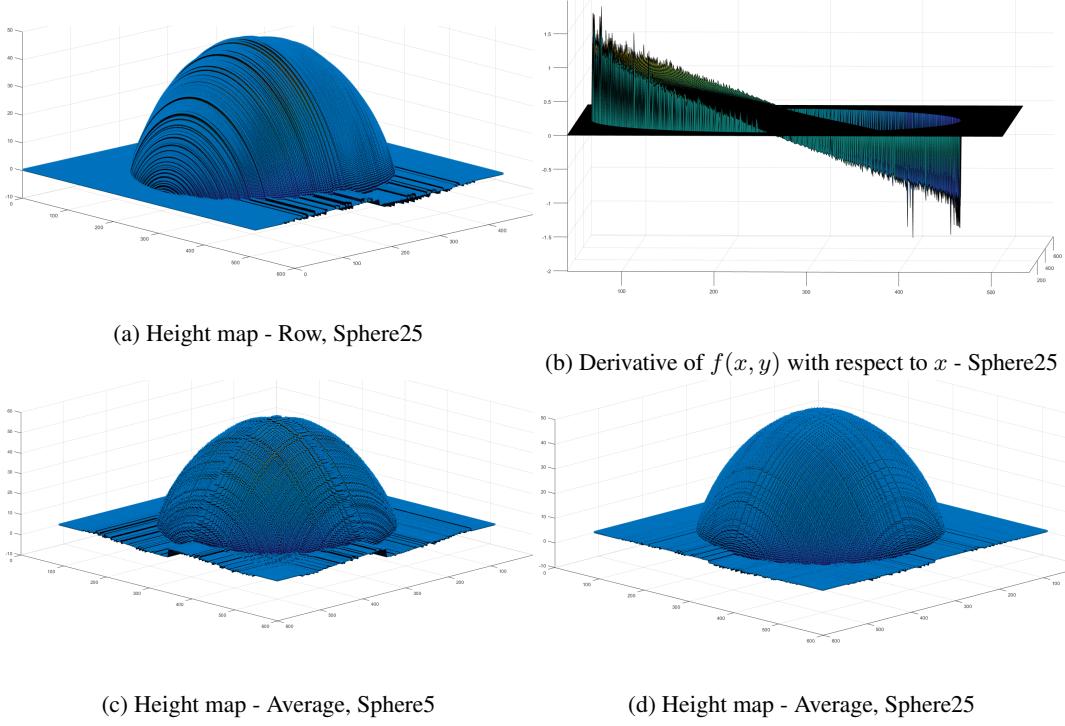


Figure 4: Shape by integration

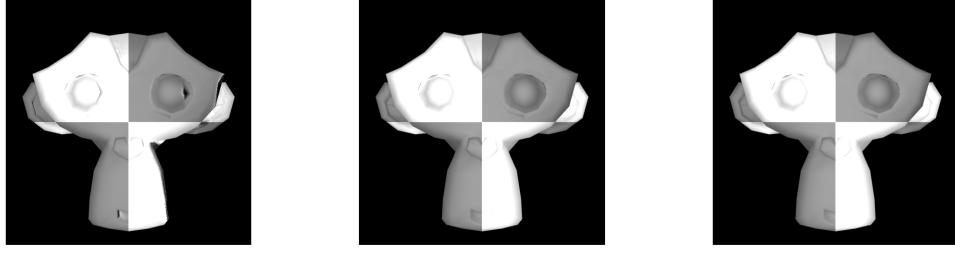


Figure 5: Monkey face albedo reconstruction for different image sets

The experiment suggests that a brute force approach might not work so well in fixing the errors. Doubling the number of images did not produce drastic improvements. However, it could be that focusing more light sources around the patches that show errors could be very beneficial. For example, with only a few images with a light source from the right, much of the artefacts in albedo reconstruction are cleared. Finally, changing the model to account for changes in ambient lighting is another option which we do not treat here.

Question 5 In order for the algorithm to work for an image with three channels, it had to be modified. We applied the same process as before, and computed the albedo and the surface normal for each pixel, independently for each channel. The albedos for the three channels are then combined to form an RGB image. In order to form the surface normal map for the image, the individual normals are summed. We can see that this approach yields decent results. Figure 6 shows the result the modified algorithm produces. In sub-figure (b) we can see albedo and the height map combined. The representation is a good representation of what the virtual object is expected to look like. Moreover, the accuracy of this algorithm is also confirmed by the first derivative of the height function with

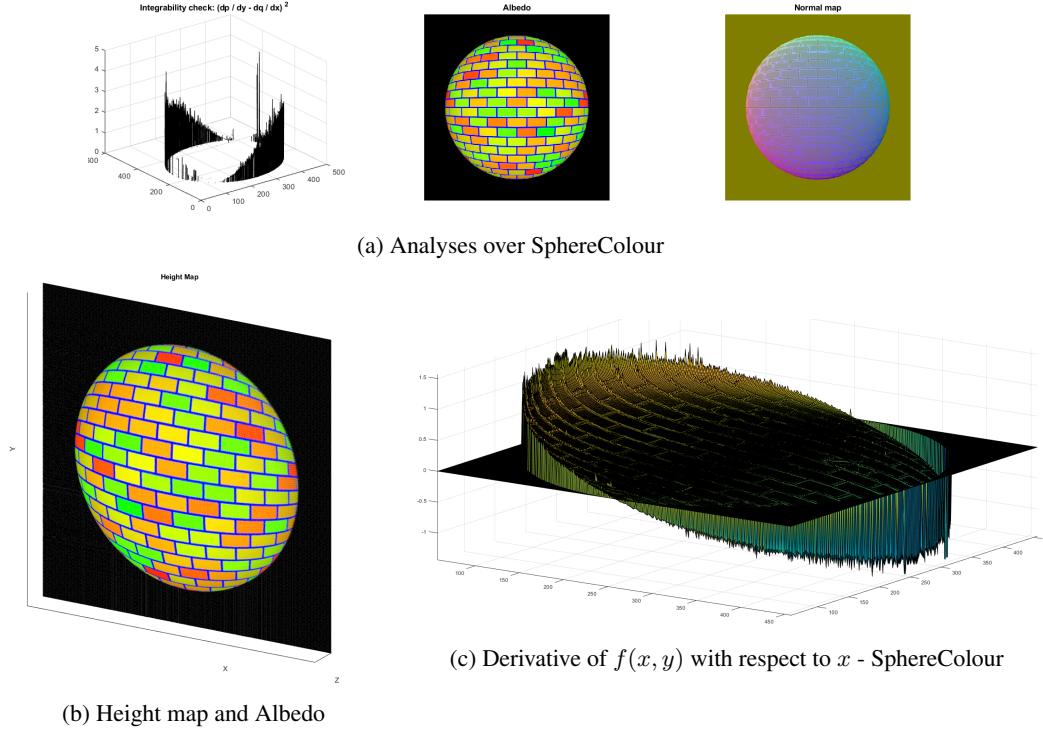


Figure 6: Coloured sphere processing

respect to x . In figure 6 (c) one can see that the height function capture fine details, such as brick spaces.

During the development of the algorithm we encountered some impediments, such as some zero pixel values. When constructing the surface normal map, the algorithm gave some invalid values(NaN) which we solved by setting them to zero. In addition to that, in the case of the MonkeyColor data-set, the green colour missed form the images, and had to deal with this issue accordingly.

Question 6 Depending on the passing orientation, we observe a general tendency for the row-major passing to outperform column-major passing 7. For example, on a row -major passing, the nose is more prominently mapped, and the unsavoury trench on the forehead is removed. The average passing in this case does not seem to be the best path to take. Although it stumps out some exacerbation around the nose of the row-major passing, it reintroduces the forehead distortion.

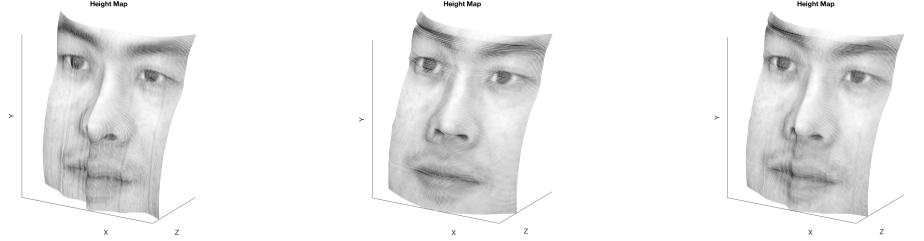
We observe that some images have a strange texture, while some are completely black. This hints that those images are violating the point light source at infinity assumption. Moreover, all images are violating the no ambient lighting assumption, since some patches, such as the nostrils seem much less of the world than other patches. Finally, the face presents specularities on the nose and eyes usually.

Finally, we try to Exclude the images which violate the assumption aforementioned, or images which are too dark, imposing unnecessary constraints on the systems of linear equations. In total we eliminate 16 images, but we do not see any noticeable difference in the height maps.

3 Colour Spaces

RGB Colour Model

With additive colours, the more you add, the brighter the colours become. On the other hand, with subtractive colours the more colours you mix, the darker the resulting colour. This is because they are absorbing light(figure 9). When you're dealing with digital screens, you're dealing with additive colours. When you're dealing with print, you're dealing with subtractive colours. The primary colours of subtraction are Red, Yellow and Blue, but for additive colours, the primary colours are others.



(a) Height map by row

(b) Height map by column

(c) Height map average

Figure 7: Yale face height map

Because the human eye has colour sensitive cells that are sensitive to Red, Green, and Blue, those are the primary colours of addition(the yellow colour is perceived by our eyes as a combination of blue and green).

It's why the computer monitor uses red, green, and blue light to create the colours on the screen, but the printer uses cyan, magenta, and yellow (black too) ink to print that same image. And since we use additive colours (RGB) to display images on monitors, it makes sense to use the same structure for the graphic card since the colours are shown without conversions.

A digital camera doesn't use a film. Instead, there is a piece of electronic equipment that captures the incoming light rays and turns them into electrical signals. This comes with some disadvantages. they are prone to errors related to the way they sense colour, such as white balance, metamerism(two colours which look different under one kind of light might look the same under another kind of light), non-visible (IR) light(digital cameras are subject to interference from IR light) and different colour definitions(every camera natively shoots into a colour space of its own).

Colour Space properties

Opponent colour space. The opponent colour space detailed in the assignment has 1 luminance component and 2 chrominance components. O_1 is the red-green channel, O_2 is the blue-yellow channel, and O_3 is the luminance channel. In figure 8c we can see that all shading is represented only in the third component, with the other two holding only chromaticity information. It is proposed that RGB does not accurately represent the human perception of colour, while opponent colour does.

Normalised RGB. Normalised RGB is an intensity-invariant colour space because all colour channels are normalised by the sum all three values. This also eliminates any shading and shadow effects, as shown in figure 8d. This provides an easy way to decompose an image to recover colour invariant features. The new image will contain only information regarding chromaticity, with all shading being neutralised.

HSV Colour Space. The Hue-Saturation-Value colour space has the advantage of being a much more intuitive space to work for humans. The ‘brightness’ of a colour is directly given as a channel named Value. Saturation reflects the amount of colour, while Hue represents the actual base colour. Apart from this, a benefit of HSV in image decomposition is that hue is invariant to shading, shadows and intensity, but also to specular reflections. Result can be seen in figure 8a.

YCbCr Colour Space. The YCbCr colour space is a colour space which again isolates a luma component, which is a non-linearly encoded light intensity. The other two channels are blue difference and red difference chroma components. This colour space is used mostly in high definition renderings. The luma component can be transmitted at high resolution / high bandwidth, while the chroma components can be subsampled, compressed, or otherwise manipulated to increase efficiency while controlling the level of error introduced in a perceptually meaningful way. Results are shown in 8b.

Grayscale. The grayscale colour space collapsed 3 values into one. This value is obtained either by taking the difference of the extremes, or averaging using a weight scheme or uniformly. The benefits of representing images in grayscale are mainly related to efficiency of storage and processing. The downside is obviously that most of the information is lost in this representation. Comparing the four

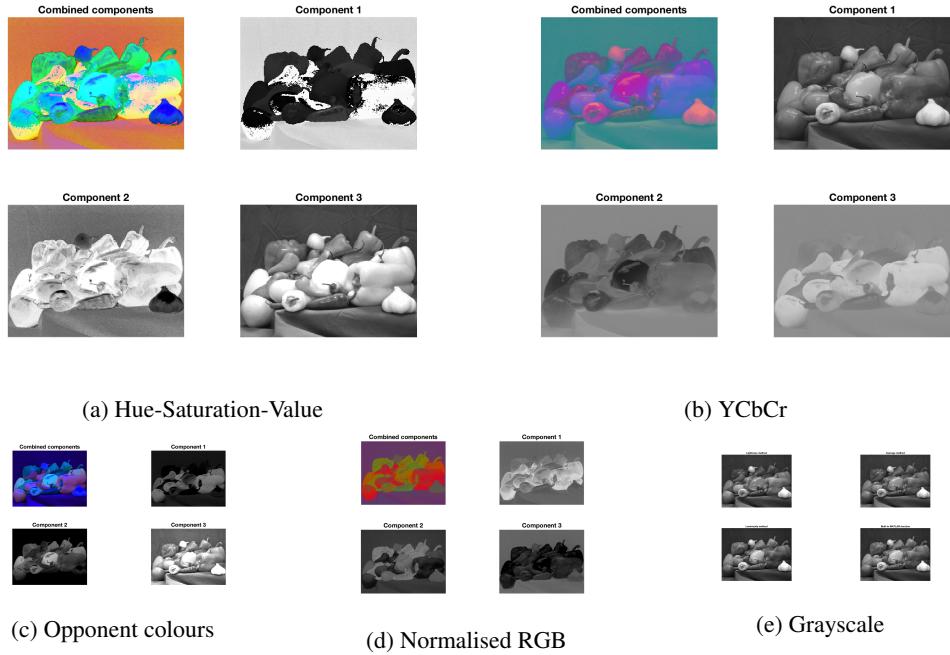


Figure 8: Colour spaces

algorithms, it can be seen that the luminosity method keeps contrasts while other methods reduce the contrast when transforming to grayscale, as shown in figure 8e.

More on Colour Spaces

CMYK Colour Space CMYK is an abbreviation for Cyan, Magenta, Yellow, Key (black). It refers to a subtractive colour model, and process itself, of colour printing. The model is referred to as subtractive because each colour subtracts brightness from white. While printing CMYK model on white paper (standard), each addition of colour cuts down on reflected light, and as each of the three colours are layered on top of the other, you will have a dark, semi-black, muddy shade. Because "black" generated is unsatisfactory, unsaturated and very dark colours are produced by using black ink instead[1].

It should be understood that your eyes have a visible spectrum that's in the billions of colours. A computer monitor can display millions, and really high quality printers can display/produce in the thousands. As such, a comparisons between RGB displays and CMYK prints can be difficult. A rough approximation of how colours will differ on these mediums can be seen in figure 10 (b). This is often problematic in printing digital media since what you can see on the screen is not what you'll get on the print. Millions of the possible displayed colours on a monitor are going to be outside the spectrum that a printer has available. Figure 10 (a) shows a comparison between the CMYK colour space and the RGB. As you can see, the CMYK model contains less colours than the RGB one. It also has to be mentioned that CMYK numbers are not unique. Different combinations of the four components can yield equal colours. It is the fourth component - black - that introduces ambiguity.

4 Intrinsic Image Decomposition

Other Intrinsic Components.

A third component that can be extracted is the shape of the image. This is given by the normal map. This can be then used to more accurately reconstruct the image instead of distributing the effects of shape over to albedo and illumination.

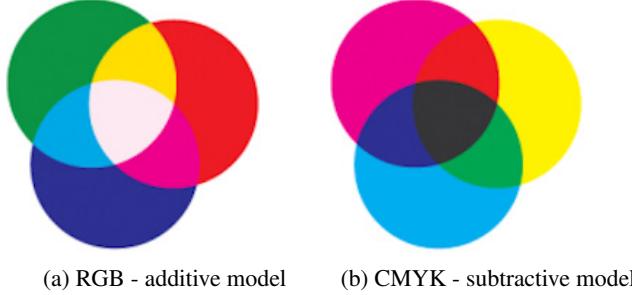


Figure 9: Integrability results

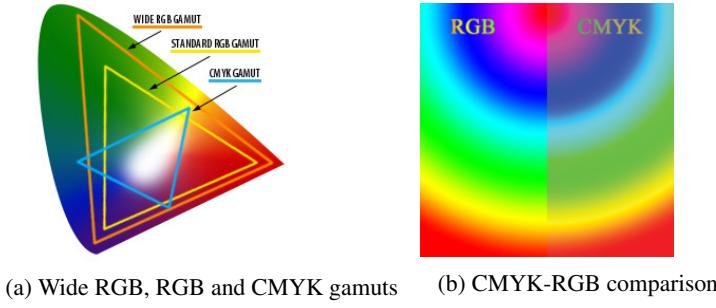


Figure 10: Integrability results

Another component is specular reflection. A classic image decomposition algorithm assumes a Lambertian surface and will only recover diffuse reflections. However, we would like to also recover specularities, less the reconstruction will never be accurate and valuable information is also lost about the direction and type of the light source. This can be achieved using Phong's empirical model for specular reflection.

$$\text{Specular Reflectance} = I \rho_s (\cos\phi)^{n_s h i n y} \quad (4)$$

The entire reflectance can then be decomposed as a linear combination of diffuse and specular reflectance.

$$I = R \cdot S = (a \cdot \text{Diffuse Reflectance} + b \cdot \text{Specular Reflectance}) \cdot S \quad (5)$$

Synthetic Images.

The problem of image decomposition is recognised as an ill-posed problem with one known, the image values, and two unknowns, reflectance and shading.

$$I(\vec{x}) = R(\vec{x}) \times S(\vec{x}) \quad (6)$$

Consequently, one needs some (partial) information about the ground truth of one in order to obtain the other. Because this ground truth is impossible to get for real images, without considerable effort, the norm is to generate synthetic images for which the reflectance and shading is known. These generated images can then be used in a similar fashion as real images, as long as they are realistic enough, for example, for training an intrinsic decomposition algorithm.

Image Formation The image reconstruction process is shown in figure 11. The original image is shown in (a), with its intrinsic decomposition in reflectance and shading in (d) and (g) respectively. As can be seen, the image has been successfully decomposed and reconstructed in (h).

Recolouring The true material colour of the ball, given in RGB format is: Red-184, Green-141 and Blue-108. Equation 6 gives us how the reflectance and the shading form the image. If we recolour

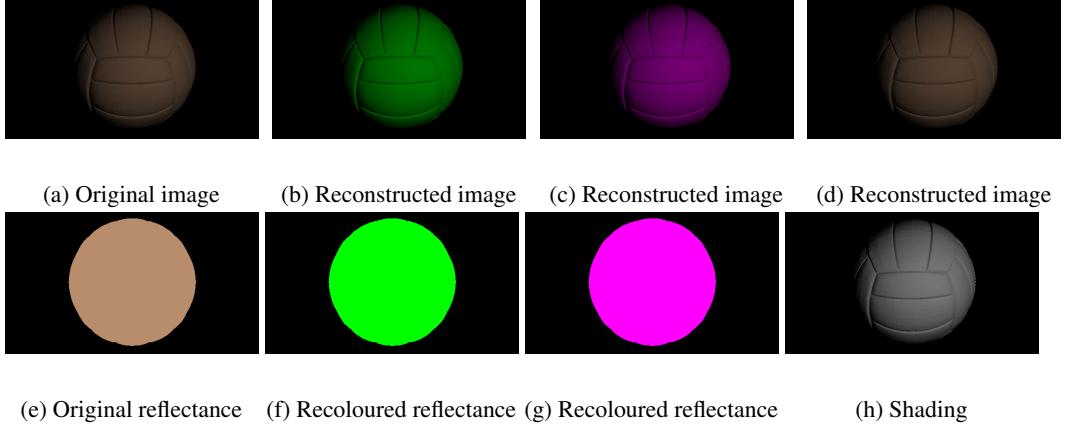


Figure 11: Recolouring

the reflectance and apply back the shading component, we get an artificially generated image that resembles with the original one, with the mention that it has a different colour(or, to be more precise, a different shade of the replacing colour). All other features remain unchanged. Figure 11 helps us visualise the mentioned process. One can notice that the object in the figures 11 (b) and (c) do not have the same vivid colours, as it is presented in figures 11 (e) and (f). This is because the information about the illumination and the surface normals from vector $S(\vec{x})$ were not lost in the process.

5 Colour Constancy

Grey-World Algorithm

Task 1 Digital cameras perceive the same object as having different colours under different lighting sources. Moreover, even the sensor of the different devices available are somehow different and might be biased. Invariance to the colour of the light source is a property that is desired and it can be approximately achieved by various colour constancy algorithms, such as grey-world, white-patch, Retinex, or ACE.

Colour constancy algorithms try to estimate the illuminant of the light source and correct the image so that it appears to be taken under a white light. Automatic-white-balance (AWB) algorithms have been incorporated in our daily devices that capture images so that the images taken look as natural as possible. To achieve that, the Grey-World Algorithm assumes that, under a white light source, the average colour in a scene is achromatic (grey, [128, 128, 128]).

Figure 12 (a) and (b) shows the Grey-World Algorithm in practice. We can observe how the red tint of the image disappeared. It can be observed the best on the colour of the house, which in the original image it has a faded pink colour, whereas in the corrected image, it looks more ‘natural’, in the sense that we are used to seeing this colour for houses.

Task 2 Although the grey-world algorithm works reasonably well in practice, it is also susceptible to yielding erroneous result when applied on an image with a large dominant colour patch. We tried to run it on an image containing many yellow Christmas lights and not surprisingly, the orange-yellow tint of the lights was diminished and the resulting colour was not the natural colour of the lights. The result can be seen in figure 12 (c) and (d). This takes place because in the original image, the average colour is Red:120.6542, Green:55.3985, Blue:6.1881, and the algorithm increases the blue and the green values so that the average colour is Red:128, Green:128, Blue:128. It can also be seen that the spots that were close to white, in the corrected image they have a cyan(blue and green) tint.

Task 3 Another simple algorithm for colour correction assumes that the brightest patch in an image generates maximum and uniform values across the RGB channels. This is referred to as the ‘white patch’, ‘scale-by-max’, or ‘max-RGB’ assumption [2]. The method finds the brightest spot in the image, computes a scaling factor in order to make that patch reflect maximally and uniformly across the three channels and scales all pixel values by that value. It is important to notice that this method



Figure 12: Gray-World Algorithm

works best when a specular highlight exists to be taken as the white patch, because patches of diffuse reflection will be chromatically biased.

A more advanced algorithm is called ACE and works in two stages by blending together the grey world and white patch assumptions [3]. The first stage, chromatic adaptation outputs an image R , where each pixel is calculated according to:

$$R_c(p) = \sum_{j \in I - \{p\}} \frac{r(I_c(p) - I_c(j))}{d(p, j)} \quad (7)$$

where $I_c(p) - I_c(j)$ is the lateral inhibition mechanism (the ability of a pixel, in this case, to reduce the response of its neighbours), $d(\cdot)$ is a distance function that controls the trade off between local and global to the inhibition mechanism, and finally $r(\cdot)$ controls the relative lightness appearances of a pixel. The function $r(\cdot)$ is chosen to be an odd function to emulate a grey-world correction, and a non-linear one to emulate a local white patch reference.

The second stage, dynamic tone reproduction, re-scales all values on each channel independently, using a sort of balance between the grey world and white patch assumption again, this time at a global level. The re-scaling is done as

$$O_c(x, y = \text{round}(127.5 + s_c R_c(x, y))) \quad (8)$$

where s_c is the slope of the segment $[(m_c, 0)m(M_c, 255)]$ using M_c as the white patch reference and the zero value in R_c as the medium grey reference point. Thus, this achieves a white-patch-grey-world globally balanced adaptation.

6 Conclusion

Overall, the experiments we ran provided interesting insights into the utility and limitations of several computer graphics algorithms. We see that photometric stereo is capable of accurately reconstructing a 3D model of simple objects, but falls short on more complex shapes. We've also seen how intrinsic image decomposition can be used to separate and manipulate components of an image independently. Finally, we have observed that colour constancy algorithms, such as the grey-world can help in some cases, but fall short on particular instances. We've also seen how to extend grey-world with more robust algorithms, such as ACE.

References

- [1] K. Sawh, “Cmyk vs. rgb and why you should care,” 2014. [Online]. Available: <https://www.slrlounge.com/cmyk-vs-rgb-care/>
- [2] A. Rizze, C. Gatta, and D. Marini, “Color correction between gray world and white patch,” *SPIE Proceedings Human Vision and Electronic Imaging VII*, vol. 4662, 2002. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.172rep=rep1type=pdf>
- [3] D. H. Foster, “Color constancy,” *Vision Research*, vol. 51, no. 7, pp. 674 – 700, 2011, vision Research 50th Anniversary Issue: Part 1. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0042698910004402>