

# Sky Detection in Images for Solar Exposure Prediction

Nuchjira Laungrungthip<sup>1</sup> Alan McKinnon<sup>1</sup> Clare Churcher<sup>1</sup> Keith Unsworth<sup>1</sup>  
laungrn2@lincoln.ac.nz mckinnon@lincoln.ac.nz churcher@lincoln.ac.nz unsworth@lincoln.ac.nz

<sup>1</sup> Applied Computing Group, P.O. Box 84 Lincoln University Canterbury, New Zealand

## ABSTRACT

This project describes a technique for segmenting the area of sky in an image from the remainder of the image. This can then be used for predicting solar exposure at a location at any time of the day on any day of the year, using a new technique which involves image processing. A series of photos is taken from a location of interest and then processed to separate the areas of sky from the rest of the image. The sunlight that will fall on the location from where the images were taken can then be calculated. Critical to the success of this technique is the image processing used to separate the sky from the rest of the image. This paper is concerned with finding a technique which can do this for images taken under different conditions.

## Categories and Subject Descriptors

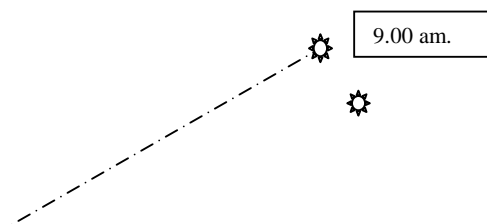
I.4.9 Applications

## Keywords

Solar exposure, skyline, image processing, image segmentation

## 1. INTRODUCTION

The sun is very useful in different situations such as drying food for agriculture, heating domestic water or powering a solar house [1]. The position of the sun in the sky changes continually during the day and seasonally throughout the year. Exposure to the sun, at a particular location at different times of the year, can be affected by obstructions such as hills, trees or buildings. Figure 1 shows the solar exposure at a particular location on a specific day at 7.00 am. and 9.00 am. The person is not exposed to the sun at 7.00 am. but he is exposed at 9.00 am.



This paper was published in the proceedings of the New Zealand Computer Science Research Student Conference 2008. Copyright is held by the author/owner(s).

NZCSRSC 2008, April 2008, Christchurch, New Zealand.



**Figure 1: The sun can be blocked by obstructions at different times of the day, or year.**

Knowing the solar exposure at a particular location would help a builder determine the position where a solar collector would get the most sunlight. It might also assist an architect who wants to set the location of windows so they receive the most sunlight in the morning.

For this project, “solar exposure at a location” is defined as the ability of a ray from the sun to intersect a particular point at any time of the day for any day of the year. Exposure may be determined for a specified time or integrated over a time period such as an hour, day, week, or month.

The concept of a device that predicts the solar exposure at a location was developed by McKinnon [2]. The following steps are involved:

1. Take a series of digital images from the location of interest in known directions.
2. Find the sky in each image using image processing techniques.
3. Determine whether a ray from the sun can reach the location by passing through the sky part of one of the images.
4. Repeat step 3 for all times of the year of interest.
5. Display the information.

A commercial device using the same principles is the Solar Access Measure Device (SAMD) [5].

A critical step is to determine which part(s) of each image correspond to sky and which do not. The processing required to do this is the subject of this paper.

## 2. OTHER APPROACHES TO IDENTIFICATION OF SKY IN DIGITAL IMAGES

This section reviews some previous approaches to identifying sky in digital images.

### 2.1 Region Extraction

Luo and Etz [3] classify sky in an image by colour value, and extract a connected component of sky colour. The colour classification uses a neural network to determine a belief value for all candidate blue sky pixels in the image. Next, the region extraction process determines an appropriate threshold for the sky colour by creating a histogram of the belief values and finding the first valley point. If a belief value of a pixel is lower

than the first valley point, the pixel is marked as a non sky pixel. If it is higher than the first valley point, the pixel is a sky pixel.

## 2.2 A Model Based Approach

Gallagher [4] uses a model based approach to classify pixels as sky or non sky. Firstly, the method of Luo and Etz [3] is used to identify pixels that represent an initial sky region. Then, a mathematical function is fitted to some of the pixels which have high sky belief values. Finally, the model is used to determine if additional pixels outside the initial sky region should be considered as part of the sky. The approach is limited to images with blue sky.

## 2.3 Sky Line Detection

The SAMD patent application [5] outlines various gradient and other approaches to detecting the skyline in the image but it is not clear how this is done in the SunEye device currently available for purchase.

## 3. THIS STUDY

This project seeks to develop a robust image processing technique for identifying the sky region of an image to enable accurate calculation of the solar exposure at a location at any time of the year.

The general approach is to determine the edges of sky regions using pixel intensity gradients<sup>1</sup>. The following steps are involved:

1. Extract a colour channel.
2. Apply the Canny edge detection algorithm [6] to determine the boundary lines in the image.
3. Apply the morphological closing algorithm [7] to close gaps in the boundaries identified by the Canny edge detector.
4. Identify which of the closed regions are sky.
5. Display and store the segmented image.

This addresses the implementation issues related to steps 1 - 3. Further work will address the question of which of the bounded regions found are sky regions.

## 4. EXPLORATION OF IMPLEMENTATION OPTIONS

In order to implement the algorithm described in section 3, it is necessary to investigate the most appropriate settings for the colour channel and other algorithm parameters to ensure that the final implementation is as robust as possible enabling it to work successfully on a wide range of images. In order to do this a set of images was selected and subjected to a series of experiments. This section describes the details of these investigations.

### 4.1 The Images

There are various factors that could affect sky detection such as the weather conditions including bright sky- no cloud, bright sky - scattered clouds, overcast sky - white clouds, and

overcast sky - grey clouds, as shown by the examples in Figure 2 which were used as the test set.



**Figure 2: Test images used (1 and 2) bright sky- no cloud, (3 and 4) bright sky - scattered clouds, (5 and 6) overcast sky - white clouds, (7 and 8) overcast sky - grey clouds.**

### 4.2 Choosing a Colour Channel

The objective of extracting a colour channel is to increase the contrast between the sky and the rest of the image and convert the 24 bits of colour to 8 bits of colour for the edge detection step. The colour channels tested were blue, red, green and greyscale [12, 13, 14], H, S, V, and Y, Cr, Cb [14], and L, a, b [18].

The following method was used to determine the colour channel best suited to separating the sky from the rest of the image.

1. From the image containing bright sky - scattered cloud, we manually selected a region containing only sky and a region that contains no sky, as shown in the Figure 3.



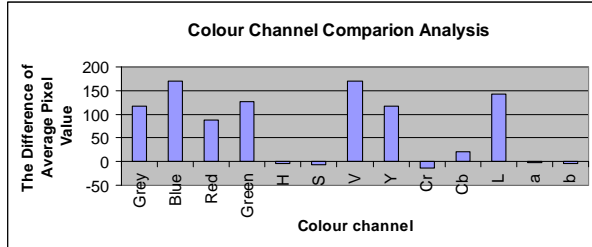
**Figure 3: The original colour image (a) is separated into sky region (b) and non sky region (c).**

<sup>1</sup> In US Patent 20070150198 the manufactures of the SunEye device [5] also refer to using row and column pixel gradients as a possible approach but do not elaborate on it.

2. All pixel values were transformed to remove the effect of the camera's gamma [16, 17] and the desired colour

channel was extracted from the sky and non sky regions. All colour channels were normalized to the range 0 to 255.

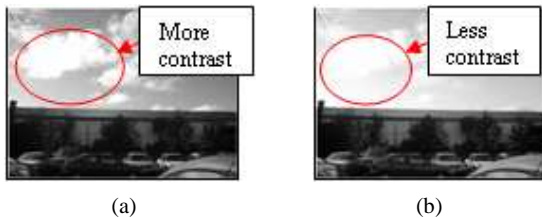
3. The average pixel value of the sky and non sky regions was calculated for each colour channel.
4. The average pixel value of the non sky region was subtracted from the average pixel value for the sky region.



**Figure 4:** Compares the average contrast value between sky and non sky parts for each different colour channel for image 4.

Figure 4 shows that Blue and V colour channels gave the highest contrast for test image 4 (bright sky – scattered clouds). When applied to all images, this process gave a contrast measure for the blue colour channel consistently greater than or equal to that for all other colour channels.

Not only does the blue colour channel increase the contrast between the sky and non sky regions of an image but it also decreases the contrast between scattered clouds and blue sky as shown in Figure 5. This reduces the likelihood that scattered clouds will be identified as regions separate from the sky in later processing.



**Figure 5:** The grayscale (a) and blue (b) colour channels for the bright sky – scattered clouds image.

### 4.3 Finding the Edges in an Image

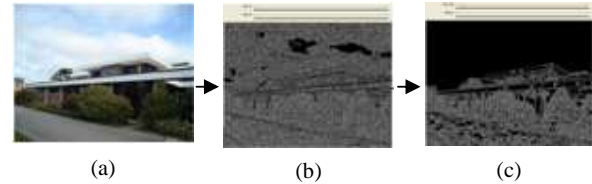
The objective of this step is to find the edges in an image so as to determine the boundaries between contrasting regions.

There are several methods to perform edge detection such as the Sobel, Roberts Cross, Prewitt, Laplace, and Canny edge detector methods [9]. The most widely accepted of these, the Canny edge detector, is used here.

The Canny operator [6] uses multiple processes to detect edges in an image. First, it smoothes an image with a Gaussian filter to reduce noise and unwanted texture and detail, then it calculates the edge gradient strength which is a measure of the change in image intensity at the edge. The edge direction is then determined and nonmaximal suppression is applied to make a thin line in the edge direction and suppress any pixel that is not an edge. Finally, hysteresis thresholding eliminates the breaking up of edge contours. Any gradient values that are higher than an upper threshold represent edge pixels and any pixel values that

connect to those edge pixels and are higher than the lower threshold, represent edge pixels as well.

The Canny edge detector may also find edges around clouds in the sky so it is necessary to adjust the thresholds to avoid those edges, as shown in Figure 6. Section 4.6 describes the results of tests carried out to find the appropriate upper and lower thresholds.



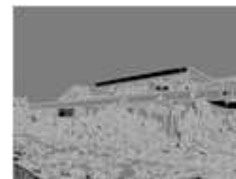
**Figure 6:** Image (a) shows the blue colour channel prior to the application of Canny edge detection. Image (b) shows the edges detected when the upper and lower thresholds are set to low values, while image (c) shows the result when the upper and lower thresholds are adjusted to avoid the edges around clouds.

### 4.4 Closing Gaps in Edges

The approach being taken to finding the sky is to find the edges surrounding all possible sky regions. Although the Canny edge detector will help to do that, it may well leave gaps in some edges. If these are not closed the area of sky may “spill” through the gap as shown in Figure 7. To counter this, the morphology closing operation [7] is applied to the binary image of edges.

The morphology closing algorithm [7] uses a dilation operation followed by an erosion operation. The dilation and erosion operations are performed by placing the middle pixel of a structure element, which is a uniform intensity pixel array (e.g. 3 x 3 or 5 x 5), over each pixel in the image and comparing the value of each structure element pixel with the pixel it overlays. If they are all the same value, we set the image pixel to 1 for the erosion algorithm and 0 for the dilation algorithm. If there are large gaps, a larger structure element will be required to fill them.

Section 4.6 describes the results of tests carried out to find the appropriate size of the structure element.



**Figure 7:** Shows what happens when a gap in the skyline is not closed. The grey colour represents the sky.

### 4.5 Identifying all the Pixels in a Region

Having determined the boundaries of all the regions in the image we can find all the pixels in any given region using the FloodFill algorithm [10]

A pixel in the region of interest is identified manually and the FloodFill algorithm then finds all pixels in that bounded region, as shown in Figure 8.



**Figure 8:** The result of the image after applying a FloodFill algorithm to the upper region in Figure 6 (c). The grey colour represents the sky.

#### 4.6 Finding Appropriate Upper and Lower Thresholds and the Structure Element Size

In order to find the boundary of any sky regions in an image, using the process described in this paper, it is necessary to determine values for the upper threshold for the Canny edge detector, the lower threshold for the Canny edge detector, and the size of the structuring element for the morphological close operation.

To make the image processing as robust as possible it is necessary to use values for these parameters so that they successfully identify the edges of the sky region(s) in a wide range of images. As there is no theoretical basis for setting these parameter values, an empirical study was undertaken to find the most appropriate settings as follows:

1. Determine the area and perimeter of the sky region in each of the sample images shown in section 4.1. The region boundary of the sky was set by manually adjusting the upper and lower thresholds and the size of the structure element until the sky boundary was seen to coincide visually with the boundary obtained by inspecting the image. This information was stored in a database table ImageInfo with imageNumber, as the primary key as well as a binary code for each image. For example, image1 has the binary code 1, image2 has the binary code 2, image 3 has the binary code 4 etc.
2. Find the area and perimeter of the sky region of each of the sample images using a range of values for the upper and lower thresholds and the size of the structure element. The area and perimeter were determined for all combinations for the following ranges:
  - upper threshold 0 to 999 increment 1
  - lower threshold 0 to 999 increment 1
  - structure element size 0 to 9 increment 1.

This resulted in the area and perimeter being determined 10,000,000 times for each of the 8 images. For each combination of settings the following information was written to the database table Information: ImageNumber, UpperThreshold, LowerThreshold, StructureElement, Area, and Perimeter.

3. Find the intersection of all the upper and lower threshold values and structure element sizes by creating a database query that finds the values of the upper threshold, lower threshold and structure element size so that the area of the sky region was within 0.3% of the value determined in step 1, and the perimeter was within 5% of the value determined in step 1. The query is shown in Figure 9.

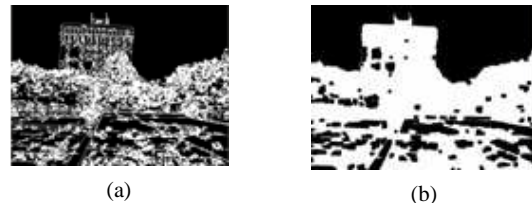
```
SELECT      i.LowerThreshold,      i.UpperThreshold,
            i.StructureElement,
            SUM( b.BinaryCode) AS sumBinary
FROM        Information AS i INNER JOIN ImageInfo
            AS b ON i.ImageNumber =
            b.imageNumber AND i.Area * (1 - 0.003)
            < b.Area AND i.Area * (1 + 0.003) >
            b.Area AND i.Perimeter * (1 - 0.05) <
            b.Perimeter AND i.Perimeter * (1 + 0.05)
            > b.Perimeter
GROUP BY    i.LowerThreshold,      i.UpperThreshold,
            i.StructureElement
```

**Figure 9:** The query used to find the upper and lower threshold values, the structure element values, and the sum of the binary code for all the images that meet the criteria.

Note that the query in Figure 9 sums the binary code for all the images in the range, so that it is possible to deduce which images do not meet the criteria. For example, if the area of the sky region from step 1 is 30,000 pixels, then the acceptable area would be in the range 29,910 to 30,090. If the summed binary code is 255 then all images met the criteria. If the total binary code is 254 all the images except image 1 met the criteria.

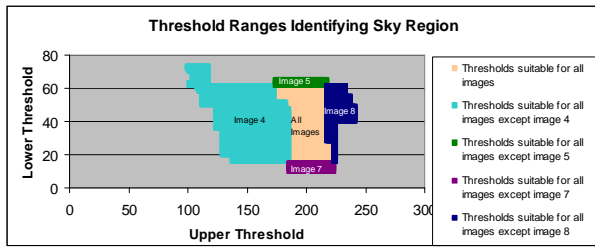
4. Find the structure element from the result of step 3. The size of the structure element, that met the criteria for the all images and had the widest range of threshold values was 3 x 3.

Figure 10 shows the result of using a 13 x 13 structure element. The area of the sky becomes smaller, by more than 0.3% of the value determined in step 1.



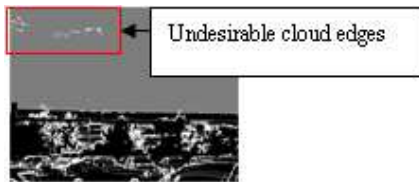
**Figure 10:** Compares images which have the same value of the upper and lower thresholds but different structure element sizes. Image (a) uses a 3 x 3 structure element while, image (b) uses a 13 x 13 structure element.

5. Find the upper and lower thresholds from step 3 by analysing the query results to show the ranges of upper and lower threshold values and structure element sizes which are suitable for all images. Figure 11, shows this information as well as the ranges that are satisfactory for all but one of the images. The upper threshold that works well in all images is in the range 186 to 217 and the lower threshold is in the range 16 to 61.



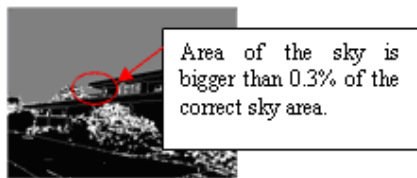
**Figure 11:** Shows the threshold ranges which gave area and perimeters of the sky region within 0.3% and 5% respectively of the correct values.

Figure 12 shows when the upper threshold equals 150 and lower threshold equals 40, these values failed to meet the criteria because cloud edges were detected which made the perimeter larger for image 4.



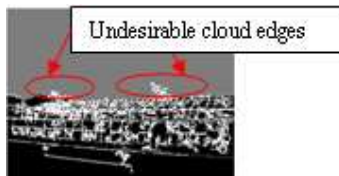
**Figure 12:** The result for image 4 when the upper threshold equals 150 and lower threshold equals 40.

Figure 13 shows when the upper threshold equals 200 and lower threshold equals 60, these values give a sky bigger than plus 0.3% of the correct area for image 5.



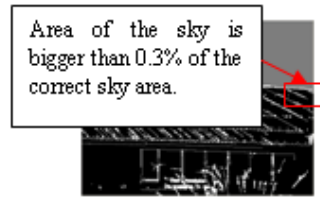
**Figure 13:** The result for image 5 when the upper threshold equals 200 and lower threshold equals 60

Figure 14 shows when the upper threshold equals 200 and lower threshold equals 10, these values produce spurious regions in the clouds which violate the perimeter criterion for image 7.



**Figure 14:** The result for image 7 when the upper threshold equals 200 and lower threshold equals 10.

Figure 15 shows when the upper threshold equals 225 and lower threshold equals 40, those values give a sky area that is more than 0.3% greater than the value determined in step 1 for image 8.



**Figure 15:** The result for image 8 when the upper threshold equals 225 and lower threshold equals 40.

## 5. DISCUSSION

The technique we have discussed successfully outlined the sky region in our sample set of images. There are other situations we have not yet fully explored which we discuss here.

### 5.1 Other Image Factors

The set of images used for this study, described in section 4.1, does not include an image for an overcast sky – black clouds, as there was not one available. However, as long as there is sufficient contrast between the sky and other objects in the image, we would expect the threshold and structure element values found here to work satisfactorily for such an image.

### 5.2 The Appropriate Upper Threshold, Lower Threshold, the Size of the Structure Element

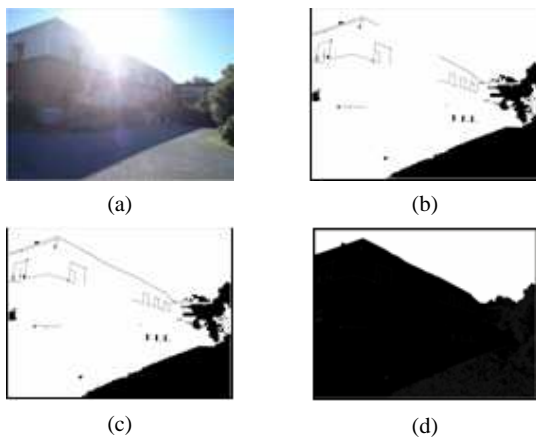
The range of upper and lower thresholds that work well for a wide range of images is 186 to 217 and 16 to 61, respectively and the size of the structure element is 3. However, the threshold values will depend on the image resolution (800 x 600 in these tests) and the particular implementation of the Canny edge detector used (OpenCV [15] in this case). The tests would need to be repeated for other situations.

### 5.3 The Area and Perimeter Criteria

The values of 0.3% for the area criterion and 5% for the perimeter criterion were selected following some initial experimentation. A more stringent area criterion was set to reflect the fact that when the sun path calculations are carried out, their accuracy will largely reflect the area of the sky region rather than the perimeter. The perimeter criterion limits factors such as spurious cloud edges as shown in Figure 12 or small deviations in shape of the boundary.

### 5.4 User Interaction

Although this work has found settings for the structure element size and threshold values that give satisfactory results across a wide range of images, there will be some situations where image processing alone will not correctly determine sky area boundaries. For example, images which are taken directly into the sun or which involve sun light reflection off very shiny surfaces will require some user interaction to identify the boundary of the sky area. Line drawing the sky boundary, painting the sky region or painting the non sky region are two of the user interaction tools that can be used, as shown in Figure 16.



**Figure 16: Demonstrates user interaction techniques. The first image (a) shows the original image (b) shows the boundaries identified by the image processing. Image (c) shows how the sky boundary could be corrected using line drawing while, image (d) shows the result of using a tool to paint the non sky region.**

## 6. CONCLUSIONS

In this paper, image processing techniques are used to identify regions in an image including the sky region. We have shown that the Canny edge detector followed by the closing morphology algorithm applied to the blue colour channel can satisfactorily find the boundaries of the sky region of an image.

We have shown that with a 3 x 3 structure element and with the upper thresholds in the range 186 and 217 and lower thresholds in the range 16 to 61, the boundaries of the sky region can be accurately determined across a wide range of images.

However, the process outlined in this paper will also identify regions in the image in addition to the sky regions. Further work is being carried out to automatically determine which of the regions are actually sky. The approach being taken is based on the brightness, colour, and relative area of the regions.

## 7. ACKNOWLEDGMENTS

The authors would like to thank the Lincoln University Image processing and Solar Radiation Masters Scholarship to support this project.

## 8. REFERENCES

- [1] Energy Information Administration. 2006. Solar – Energy from the sun. DOI=<http://www.eia.doe.gov/kids/energyfacts/sources/renewable/solar.html>
- [2] McKinnon, A.E. 2007. Estimate Solar Exposure at a Particular Location – Demonstration of Concept. Unpublished report, Applied Computing Group, Lincoln University.
- [3] Luo, J. and Etz, S. 2003. Method For Detecting Sky In Images. United States Patent Application Publication. US 6,504,951 B1.
- [4] Gallagher, A. C., Luo, J. and Hao, Wei. 2005. Detection of sky in digital color images. United States Patent Application Publication. US 2005/0147298 A1.

- [5] MacDonald, W. S. 2007. Solar Access Measurement Device. United State Patent Application Publication. US 2007/0150198 A1.
- [6] Green, B. 2002. Canny Edge detection Tutorial. DOI=[http://www.pages.drexel.edu/~weg22/can\\_tut.html](http://www.pages.drexel.edu/~weg22/can_tut.html)
- [7] Fisher, R., Perkin, S., Walker, A. and Wolfart, E. 2003. Closing. DOI=<http://homepages.inf.ed.ac.uk/rbf/HIPR2/close.htm>
- [8] Fisher, R., Perkin, S., Walker, A. and Wolfart, E. 2003. Feature Detectors. DOI=<http://homepages.inf.ed.ac.uk/rbf/HIPR2/featops.htm>
- [9] Roushdy, M. 2006. Comparative Study of Edge Detection Algorithms Applying on the Grayscale Noisy Image Using Morphological Filter. GVIP Journal, 6, 4.
- [10] Vandevenne, L. 2004. Flood Fill. Load's Computer Graphics Tutorial. DOI=<http://student.kuleuven.be/~m0216922/CG/floodfill.html>
- [11] Fisher, R., Perkin, S., Walker, A. and Wolfart, E. 2003. dilation. DOI=<http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm>
- [12] Fisher, R., Perkin, S., Walker, A. and Wolfart, E. 2003. Grayscale Images. DOI=<http://homepages.inf.ed.ac.uk/rbf/HIPR2/grvimage.htm>
- [13] BobPowell.net. How to convert a colour image to grayscale. DOI = <http://www.bobpowell.net/grayscale.htm>
- [14] Baker, H. 2004, Computer Graphics. Pearson Prentice Hall. US
- [15] Intel Corporation. Open Source Computer Vision Library. DOI = <http://www.intel.com/technology/computing/opencv/>
- [16] Bourke, P. 1998. What is gamma correction?. DOI = [http://ozviz.wasp.uwa.edu.au/~pbourke/texture\\_colour/gamma/](http://ozviz.wasp.uwa.edu.au/~pbourke/texture_colour/gamma/)
- [17] Stokes, M., Anderson, M., Chandrasekar, S., and Motta, R. A 1996. A Standard Default Color Space for the Internet – sRGB. DOI = <http://www.w3.org/Graphics/Color/sRGB>
- [18] Hunter Associates Laboratory, Inc. 1996 CLE L\*a\*b Color Scale. DOI = [http://www.hunterlab.com/appnotes/an07\\_96a.pdf](http://www.hunterlab.com/appnotes/an07_96a.pdf)