Edge-Based Detection of Sky Regions in Images for Solar Exposure Prediction

N. Laungrungthip, A.E. McKinnon, C.D. Churcher, K. Unsworth
Applied Computing Group, Lincoln University, Canterbury, New Zealand.

Email: mckinnon@lincoln.ac.nz

Abstract
A device for predicting the solar exposure at a location operates by gathering image data from that location with a known camera orientation. The image data is then processed to identify the sky regions and the solar exposure is predicted using a standard sun path model and tracing the rays from the sun through the processed images.

Critical to the success of this technique is the image processing used to separate the sky from the rest of the image. This work is concerned with developing a technique which can do this for images taken under different weather conditions. The general approach to separate the sky from the rest of the image is to use the Canny edge detector and the morphology closing algorithm to find the regions in the image. The brightness and area of each region are then used to determine which regions are sky. The FloodFill algorithm is applied to identify all pixels in each sky region.

Keywords: Solar exposure, skyline, image processing, image segmentation.

1 Introduction

Being able to predict the exposure to the sun at a particular site can be very useful in situations such as 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. The exposure to the sun at a particular location at different times of the year, can also be affected by obstructions such as hills, trees or buildings.

Knowing the solar exposure at a particular location would help an architect to design an energy efficient house and also help a builder determine the position where a solar collector would get the most sunlight.

The solar exposure at a location may be of interest for a specified time or integrated over a time period such as an hour, day, week, or month.

A number of devices have been proposed [2, 5, 13] based on the following principles:
1. Take a series of digital images from the location of interest with the camera pointing known directions.
2. Find the sky in each image using image processing techniques.
3. Use a standard sun path model [14] to calculate the azimuth and elevation of the sun at any time(s) of the year.
4. Determine whether a ray from the sun can reach the location by passing through the sky part of one of the images, as shown in Figure 1.
5. Display the information.

This paper is concerned with the image processing to identify the sky regions in an image under as wide a range of conditions as possible.

Figure 1: A ray from the sun passes through the sky part of one of the images.

2 Other Approaches to Identification of Sky in Digital Images

Luo and Etz [3] determine sky areas in an image by investigating colour values, and extracting regions of appropriate 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.

Gallagher et al. [4] use 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. This 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.

The SAMD patent application [5] outlines various gradient and other approaches for 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 paper describes the development of 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 the sky regions. The following steps are involved:

1. Extract a colour plane.
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.
4. Identify which of the enclosed regions represent sky.
5. Display the segmented image.

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 these investigations.

4.1 The Images

Different weather conditions can have an effect on the ability to determine the sky regions. For example each of these conditions can pose different challenges: bright sky- no cloud, bright sky – scattered clouds, overcast sky – white clouds, and overcast sky – grey clouds. The set of images in Figure 2 were used as the test set.

Figure 2: Test images: (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.

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

1. On each test image, a rectangular region containing only sky and a rectangular containing no sky were manually selected, as shown for test image 4 in Figure 3.

Figure 3: The original test image 4 (a), sky region (b) and non-sky region (c).

2. 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 values of the sky and non-sky regions were calculated for the extracted colour channel.

4. The average pixel value of the non-sky region was subtracted from the average pixel value for the sky region and displayed.
Figure 4 shows that the Z, Blue and V colour channels gave the highest contrast for test image 4 (bright sky – scattered clouds). When applied to all images, the blue colour channel was consistently among the three channels that give the highest contrast across all test images and was therefore used in subsequent processing.

![Figure 4: The difference of the average value of the sky and non-sky regions for each different colour channel in image 4.](image)

### 4.3 Finding the Regions in an Image

The objective of this step is to find the edges of all the regions 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 [10]. The most widely accepted of these, the Canny edge detector, is used here.

The Canny edge detector may find edges around clouds in the sky. However we can adjust the upper and lower thresholds in the algorithm to avoid those edges. The results of using different thresholds are shown in Figure 5.

![Figure 5: 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.](image)

### 4.4 Closing Gaps in Edges

The approach 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 leave gaps in some edges. If these are not closed the area of sky may “spill” through the gap as shown in Figure 6. 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 use a structure element (e.g. 3 x 3 or 5 x 5).

![Figure 6: The effect of a gap in the skyline. The grey colour represents the sky.](image)

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 [11].

### 4.5 Selecting Image Processing Parameters

In order to determine the sky regions in an image, it is necessary to determine values for the upper threshold and the lower threshold for the Canny edge detector, the size of the structuring element for the morphological closing algorithm, a FloodFill starting pixel for the FloodFill algorithm, and which of the bounded regions are sky.

To make the image processing as robust as possible it is necessary to use values for the threshold settings and the size of the structuring element so that they successfully identify the sky regions in a wide range of images. As there is no theoretical basis for setting these parameters, an empirical study was undertaken to find the most appropriate settings as described in the following sections.

### 4.6 Determining edge detector upper and lower thresholds and the structure element size

The following approach was used.

1. Manually adjust the thresholds for each test image using the blue colour channel, until the sky boundary was seen to coincide visually with the boundary obtained by inspecting the image. The sky region was then identified manually and its area and perimeter were determined. We refer to these as the actual area (AA) and perimeter (PA) in the next steps.

2. Calculate the area (CA) and perimeter (CP) of the sky region for each test image when the upper and lower thresholds and size of structuring element are varied. The lower thresholds ranged from 1 to 1000, the upper thresholds from 1 to 1000, and the structure element from 3 x 3 to 21 x 21 in steps of 2 in each direction.
3. For each set of parameters tested, the maximum fractional error (EA) between the calculated (CA) and actual (AA) areas was determined across all the images according to equation (1).

\[ EA = \max_i \left( \frac{|CA_i - AA_i|}{AA_i} \right) \quad \{i = 1...n\} \quad (1) \]

where \( n \) is the number of images.

The smallest value of EA that could be achieved was 0.3% of the actual area.

4. Similarly for the maximum fractional error (EP) in the perimeter, according to equation (2). The EA was kept at 0.3% for this step. The smallest value of EA that allowed all images to be represented was 5% of the actual perimeter (AP).

\[ EP = \max_i \left( \frac{|CP_i - AP_i|}{AP_i} \right) \quad \{i = 1...n\} \quad (2) \]

5. The ranges of parameters which gave on \( EA \leq 0.3\% \) and on \( EP \leq 5\% \) were determined.

The smallest value of EA (0.3%) and EP (5%) both occurred only with a structure element size of 3×3.

The ranges of upper and lower threshold values for the smallest area and perimeter tolerances for all images are 186 to 217 and 16 to 61 respectively, as shown by the "All Images" region in Figure 7. Figure 7 also shows the ranges for the lower and upper threshold values which met the smallest area and perimeter tolerances criteria in all but one of the test images.

The following describes why some threshold values failed to meet the minimum criteria for images 4, 5, 7 and 8 as shown in Figures 8 and 9.

Figure 8(a) shows the result for image 4 when the upper threshold is 150 and lower threshold is 40 and Figure 8(b) shows the result for image 7 when the upper threshold is 200 and lower threshold is 10. These values produce spurious edges in the clouds which violate the perimeter criterion.

Figure 9(a) shows the result for image 5 when the upper threshold equals 200 and lower threshold equals 65 and Figure 9(b) shows the result for image 8 when the upper threshold equals 225 and lower threshold equals 40. These values give a sky area more than 0.3% larger than the correct area in both images.

4.7 Determining sky regions automatically

In section 4.6 the ranges of upper and lower thresholds, and the structure element size were determined to identify the boundaries of all the region in an image. However, in that analysis those bounded regions corresponding to sky were manually identified. This section describes the initial development of an algorithm to automatically determine which of the identified bounded regions are actually sky. The algorithm is based on the brightness of the pixels and the area of the regions.

For a region to be a candidate sky region it must contain one or more pixels whose value is greater than a brightness threshold. The brightness threshold selected is based on an image histogram as shown in Figure 10.
The algorithm to determine which of the candidate regions are sky is based on the size of the region. The basic idea is that possible sky regions are ranked by area and each is considered in turn to see if it should be classified as a sky region. The criterion chosen is that the region under consideration (in ranked order) should have an area greater than a specified area threshold of the accumulated sky area thus far.

The correct sky regions for all of the test images were identified only if the brightness threshold was set where 20% of the pixels were brighter than the threshold values and the area threshold was set at 50%. Although the values of 20% for the brightness threshold and 50% for the area criterion selected as a result of these tests work for all the test images, they may not be successful for other images. An example which would probably fail is where the image consists of a number of small bright areas of sky such as would be obtained if the image was taken through the branches of a tree. Further testing and revision of the algorithm is required to address these cases.

4.8 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 sunlight reflection off very shiny surfaces will require some user interaction to identify the boundary of the sky area. User interaction tools have been developed to allow the user to draw the sky boundary, paint the sky or the non-sky region.

4.9 Displaying and storing the segmented image

After identification of the sky region, the image is converted to binary form with white used to represent sky pixels and black to represent non-sky pixels as shown in Figure 11. This image is stored for subsequent processing to enable prediction of solar exposure.

![Figure 11: The final image after identifying the sky region. The white colour represents the sky. The black frame shown around it is not part of the image.](image)

5 Discussion

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

5.1 Camera Settings

In some image processing situations, it may be appropriate to control the camera's settings to optimise the image so that subsequent image processing is more likely to succeed. However, images to predict solar exposure will be taken in a wide range of conditions and it is therefore inappropriate to control individual camera settings. Rather, the approach that has been taken is to make the image processing as robust as possible by ensuring that it works successfully on a range of images, such as those in the test set, using the camera's automatic settings.

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 the test images is 186 to 217 and 16 to 61 respectively, and the size of the structure element is 3. However, the threshold values depend on the image resolution (800 × 600 in these tests) and the particular implementation of the Canny edge detector used (OpenCV [12] in this case). The tests would need to be repeated for other situations.

When the results of this investigation are applied to a new image, only a single value for each of the upper and lower thresholds needs to be specified. All the values in the range remove unwanted edges in the sky area. There will be many edges remaining in the non-sky area and the different upper and lower thresholds will affect the regions that are bounded by these edges. The lower values of both thresholds result in smaller bounded regions. These smaller areas are less likely to be mistakenly identified as sky in subsequent processing. If low values of the thresholds are used, more non-sky regions are found, but the area of each region is smaller. Smaller regions are less likely to be large enough to be classified as sky regions using the technique discussed in section 5.2. Therefore, it is appropriate to use the lowest values of the upper and lower threshold ranges, 186 and 16 respectively.

5.3 Other Image Factors

The set of images used for this study, Figure 1, does not include an image for an overcast sky with black clouds. 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.
6 Conclusions

In this paper, image processing techniques are used to identify the sky regions in an image. 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.

Although, the brightness and area criteria are used to determine which of the identified regions are sky, another situation where the bounded region is incorrectly identified is where the bottom part of the image is identified as sky. These incorrectly identified regions may not affect the solar exposure calculation because the sun path never passes through that area. However, it might be possible to improve the identification of sky regions by using the location of the region as a guide, i.e. regions at the bottom of an image are unlikely to be sky.

Although the technique developed in this paper is shown to work well with a range of sky conditions, it has not been tested with a large number of images. However, provided an image exhibits contrast similar to the test images it is likely the algorithm will successfully identify the sky regions.

Although the approach described in this paper should enable the sky region to be identified in a wide range of sky conditions, there will be some situations where image processing alone does not correctly determine sky area boundaries, for example, if an image is taken directly into the sun or sunlight reflects off very shiny surfaces. These situations will require some user interaction to identify the boundaries of the sky area. Thus we allow the user to draw the sky boundary and manually identify the sky or non-sky region.

The techniques developed in this paper work successfully across a wide range of sky conditions and can be readily combined with software which calculates the sun path and traces the path of a ray through the processed images to predict solar exposure at a location at any time of the day for any day of the year.

7 Acknowledgements

Nuchjira Laungrungthip received support from the Lincoln University Image Processing and Solar Radiation Masters Scholarship while carrying out this work.

8 References