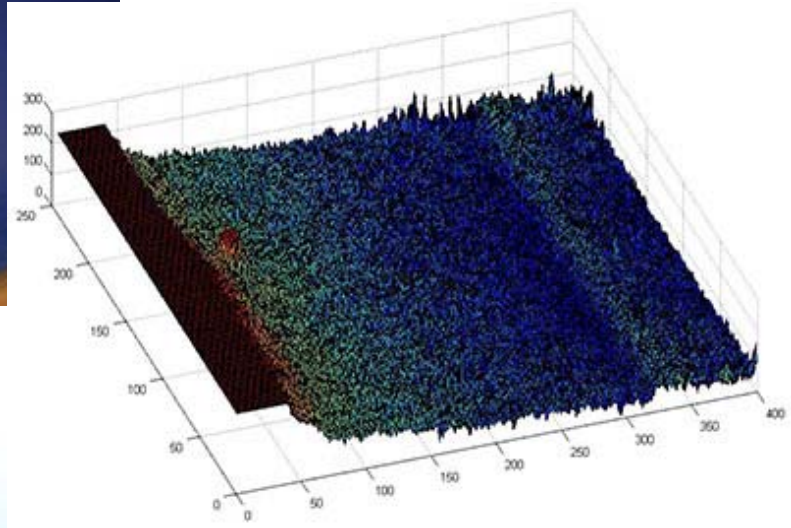
# Automatic Halo Recognition from Photographs

*Thoughts of feature extraction possibilities in halo images*




Jukka Ruoskanen

# Outline

- Introduction
- Test photos
- Processing steps
- Data
- Features
- Discussion

# Introduction

- Aurora researchers have developed / are developing pattern recognition algorithms for their automatic cameras in order to
  - detect the presence of an aurora display
  - determine the duration of a display
  - classify the activity of a display
- An automatic halo camera system would benefit from such an algorithm as well
  - tasks would be similar: presence and duration of a halo display, recognition of halo forms etc.

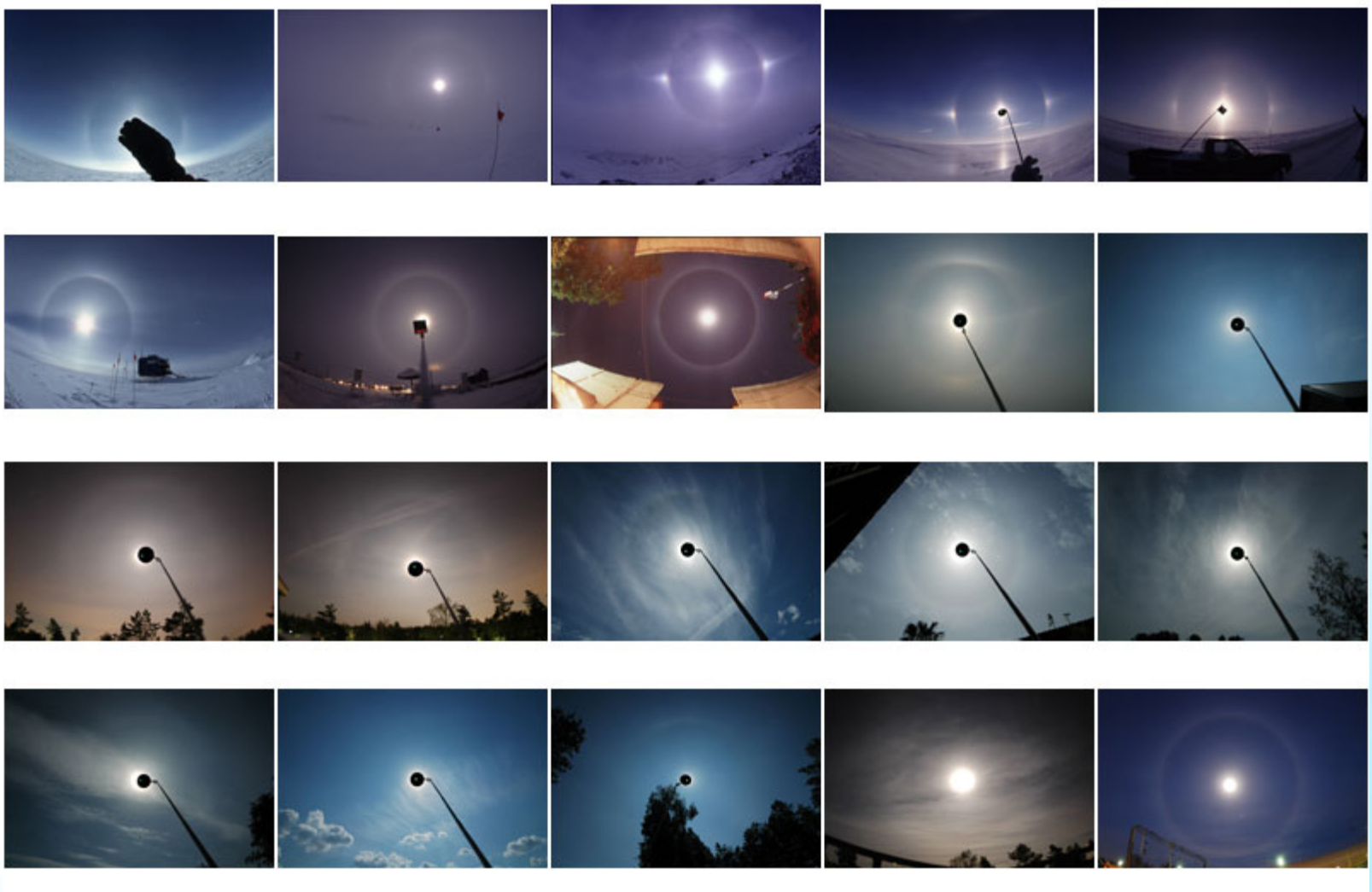
# Introduction

- Features that indicate the presence of a halo in the vicinity of 22° from Sun/Moon have been studied in Matlab environment
- The image processing steps that are needed for feature extraction are outlined and a preliminary set of promising features are presented

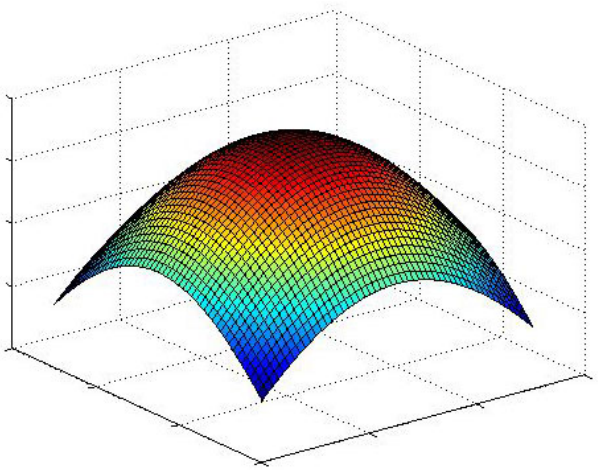
# Test photos

- A set of test photos was selected
  - 8 bit amplitude resolution digital RGB images (jpeg-format)
  - Some were scanned and some were captured with digital SLR camera
  - Lenses were 16mm (full frame and Nikon DX) and 12mm (Nikon DX)
  - The set includes day- and nighttime halos, weak and strong halos and homogeneous as well as cloudy backgrounds
  - Image size 1280 x 851 px

# Test photos



# Processing steps



- In order to enhance contrast, especially in case of weak halos, an unsharp mask filter is applied
  - first step is to blur a copy of the image with a gaussian low pass filter using a convolution kernel 51 x 51 pixels in size with the Gaussian std being 40 px.

# Processing steps

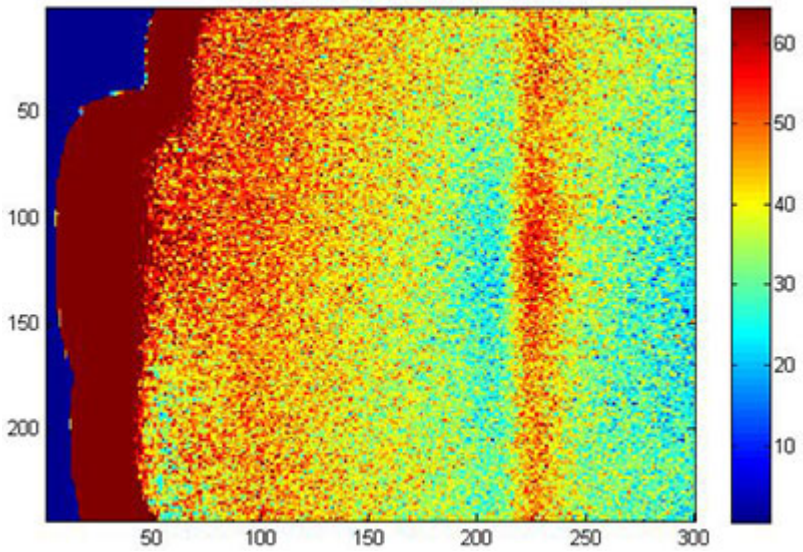
- Next and final step is to subtract the blurred copy ($I_B$) from the original ($I$):

  $I_{USM} = a * I + (1-a) * I_B$, $a > 1$
- When blurring is performed using the values presented in the previous slide and the value of $a$ is selected to be 10, the filtering has the same effect as Photoshop usm (500,40,0)

# Processing steps

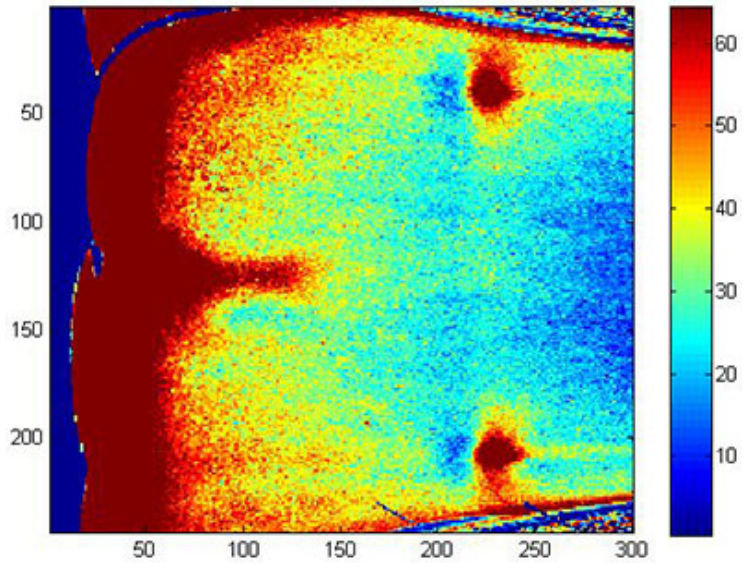
- Only a small amount of all image data is selected for use
- Lines, that define the pixels from where the data is collected, are drawn from the Sun to a distance of ~30° from it.
  - Sample lines are radial and are separated by 1°
  - 250° sector is considered
  - Data from each colour channel is collected separately

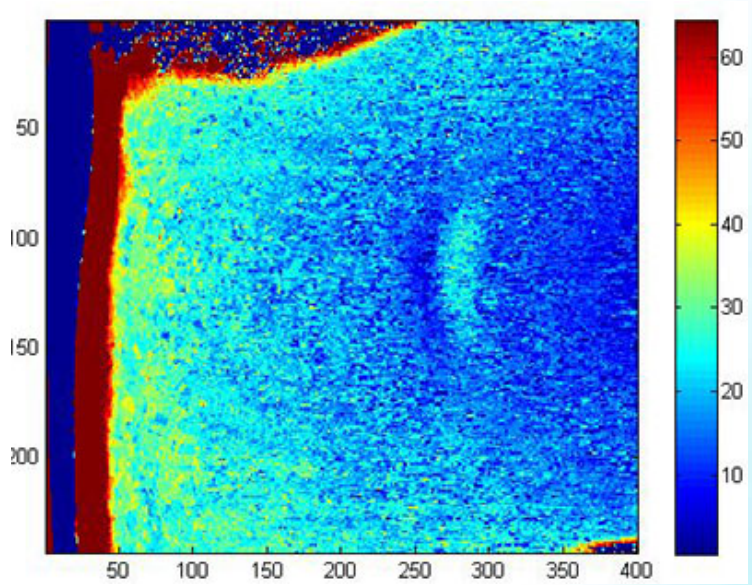
# Data




- Now we have data ready for inspection
- Here is an example of red channel data
- Data has a lot of noise in it, so averaging is needed to obtain meaningful features

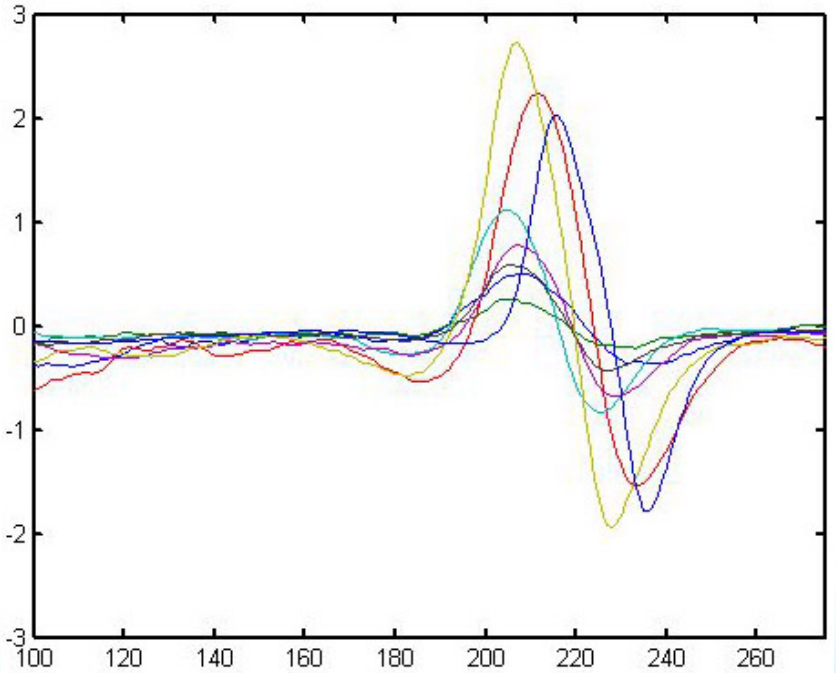
# Data

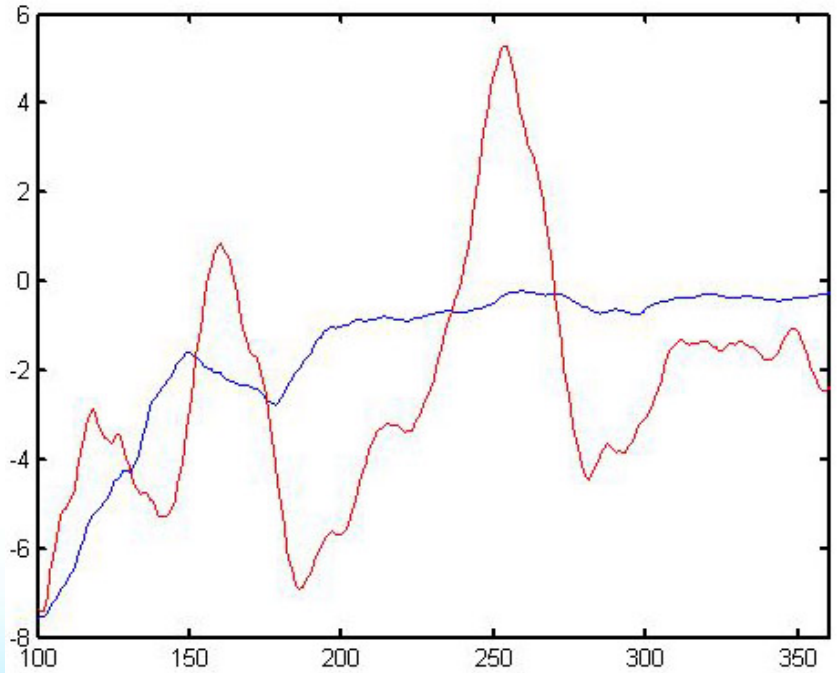

50
100
150
200
50
100
150
200
250
300
60
50
40
30
20
10


50
100
150
200
50
100
150
200
250
300
350
400
60
50
40
30
20
10

# Features

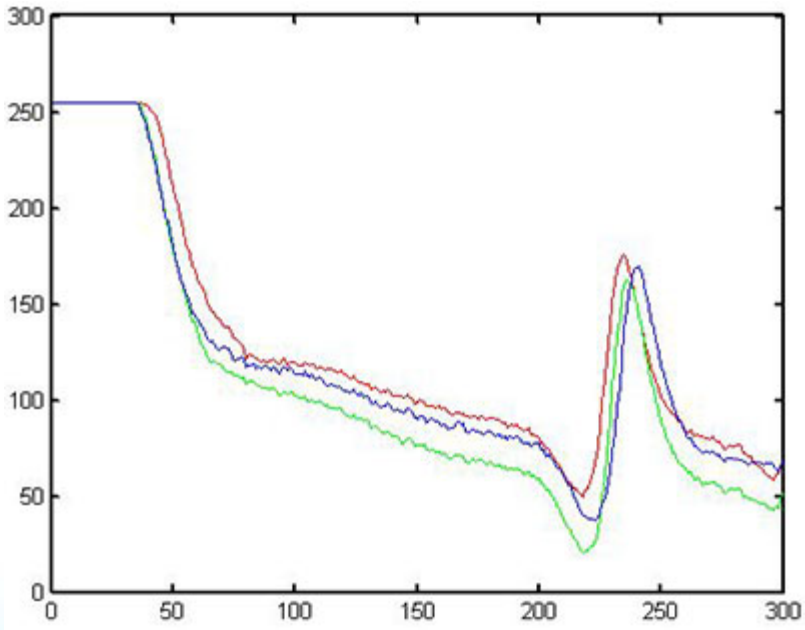
- Derivative of an averaged dataset
  - In 19 cases out of 21 the value of the derivative is > 0 at halo location – seems to be a good feature

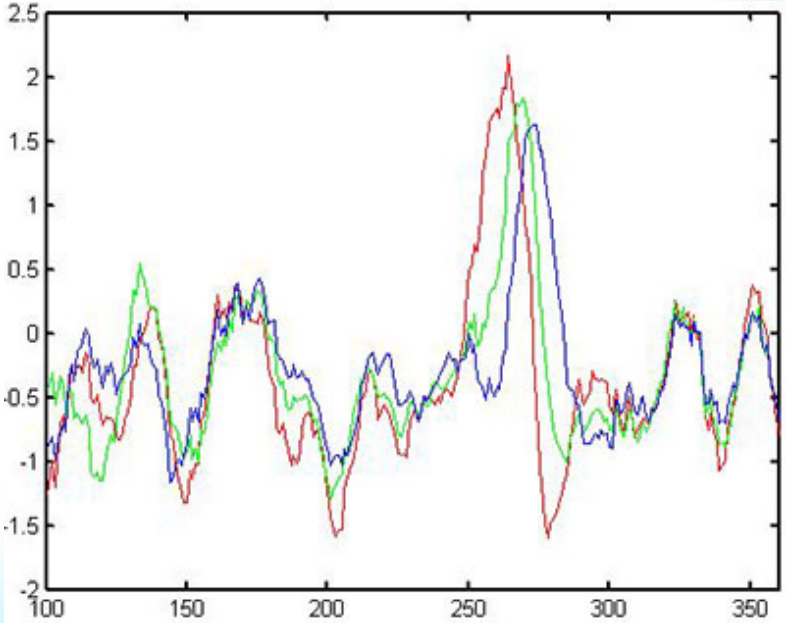


# Features

- Derivative can exceed 0 because of clouds or other elements
  - Solution is to study the three colour channels



# Discussion

- The derivative method fails in two of the test photos and the three colour channels do not bring the halos above threshold either
- The two cases are:

# Discussion

- In case the halo is very local and faint (solitary 23° upper plate arc for example) the presented techniques fail
  - if the angular data is divided into sectors which are considered separately, the possibility to detect ”short” halos or faint halo patches is increased
  - if stacking method is utilized, the signal-to-noise ratio is improved, which in its part makes detecting faint halos easier

# Discussion

- The presented features provide good results for well defined halo displays
  - in case a true pattern recognition algorithm would be used the number of independent features should be larger
  - classification of different haloforms is possible
    - reliability is increased by integration over time (stacking)
    - the mapping function of the optical device should be known