Climate Change and The Antarctic Sea Ice Extent – Visual to Statistics

ELEC 221: Signals and Systems – Winter 2016 Term 1

October 25, 2016

1 Overview

Despite the fact that the Arctic (at the Earth's North Pole) and the Antarctic (at the Earth's South Pole) are both cold and remote, they are quite different. Polar bears, for example, live only in the Arctic, while penguins live solely in the Antarctic[1]. Another probably less known difference is the way in which the sea ice at both regions respond to climate change. While global warming is well known as the major cause of ice loss in the Arctic, much less is known about its impact on ice extent at the Antarctic.

In this project, you will use MATLAB, in conjunction with some basic image[2] processing techniques, in order to extract information about ice extent at the Antarctic from a video file provided by NASA. This video includes a visual illustration of ice extent at the Antarctic in a six-month duration. The main idea of the project is to convert the visual illustration into statistics, i.e., into numbers and plots that indicate the change in ice extent during the same duration. You will begin with extracting still frames, i.e., images, from the video file. Such frames will be converted to black and white images from which the area covered by ice can be estimated by counting the number of white pixels. Finally, you will compare your results with datasets provided by the National Snow and Ice Data Center (NSIDC) at the University of Colorado, Boulder.

[1] Strictly speaking, they still can live together in the zoo.
[2] Recall that an image is a two-dimensional signal.

1.1 Requirements

You will need a computer with access to the Internet and MATLAB with the *Image Processing* Toolbox.
UBC Applied Science students can download MATLAB for free by following the links at `http://students.engineering.ubc.ca/success/software/`

1.2 Instructions

- This project must be done individually. Do **NOT** share your answers, figures, code, or report with anyone.
- You will submit a hardcopy report printed on letter-sized paper with a cover page that includes your name and UBC student number.
- The contents of the report will be **your answers to the questions** labeled **Q0**, **Q1**, ..., **Q14**, as well as **printouts of the figures** labeled **Fig. 1**, **Fig. 2**, ..., **Fig. 8**.
- The number of marks assigned to each question or figure is provided (the total is 100).
- Your answers to the questions should be sorted in the order in which they are listed.
- You should also submit a printout of your MATLAB code. Make sure that your code is properly commented.
- In order to be graded, your report must be **submitted before or during the last lecture of the course**.

2 Tasks

Task 1 Frames Extraction

Before you start, see Section 3 for a brief description of the MATLAB functions utilized in this task.

- Download the video file titled *"The Arctic and the Antarctic Respond in Opposite Ways"*, by *NASA's Goddard Space Flight Center*, from the following link onto your computer.
  `https://www.youtube.com/watch?v=J_WWXGGWZBE`

  You can use the YouTube online downloader available at
  `http://youtubemultidownloader.com`

Choose the download format as **MP4** along with a resolution of **720p**.

- Create a "VideoReader" object in order to read the video file into MATLAB.

Q0 What is the total number of frames in the video? (2 pts)
What is the frame rate? (2 pts)

- Read all the frames from the video file using the function "readFrame".
- Save each frame into an image file with **PNG** format using the function "imwrite".
- In this project, we are only interested in 12 seconds of the video; those between **0:45** and **0:56** which show the Antarctic sea ice over the course of six months in the fall and winter[3] of 2014. Specifically, the first frame should be the one stamped with the date "Mar 21, 2014" on the top left, and the last frame would be the one stamped with the date "Sep 19, 2014". You can discard the frames before and after the duration of interest.

Q1 What is the size (number of rows, columns, ...) of the data array representing each frame? (2 pts)
What is the height and width (in terms of number of pixels) in each frame? (2 pts)
What is the number of bits per pixel in a true color (RGB) image? (2 pts)

Task 2 Conversion to Grayscale

- Convert each RGB frame to grayscale.

Q2 What is the number of bits per pixel in a grayscale image? (2 pts)

Q3 Explain briefly, and possibly aided by equations, how an RGB image is converted to grayscale. (4 pts)
(If you use a MATLAB built-in function, you still need to state which one you use and briefly explain how it works.)

Fig. 1 Use MATLAB's "imshow" function to display the frame stamped with the date "April X, 2014", in grayscale, where X is given by

$$X = mod(Y, 30) + 1,$$

Y is the last two digits of your UBC student number, and *mod* is the remainder after division (the modulo operation). (5 pts)

For example, if your UBC student number is 12345678, then $Y = 78$, $X = mod(78, 30) + 1 = 19$, and you should plot the frame stamped with the date April 19, 2014.

[3] Note that seasons are opposite between the Southern and Northern Hemispheres.

Q4 Explain why conversion to grayscale is a logical step before conversion to black and white. (2 pts)

Task 3 Conversion to Black and White

- Convert grayscale images to binary ones, i.e., black and white images.

Q5 Explain briefly how conversion to black and white can be performed. Feel free to use functions provided by MATLAB, but you should state which one you use and briefly explain how it works. (4 pts)

Fig. 2 Use the"imshow" function to display the frame stamped with the date "April X, 2014" in black and white. (5 pts)

Task 4 Masking out the Date Stamp

The area covered by ice will be measured in terms of the **number of white pixels** in each black and white frame. Since the date stamp at the top left of each frame is also provided in white font, it should be removed via an appropriate mask before counting the number of white pixels.

Q6 Explain your method to remove the date stamp from the top left of each frame. (3 pts)

Fig. 3 Use the "imshow" function to display the frame from Task 3 after masking out the date stamp. (3 pts)

Task 5 Spatial Filtering

Since we are interested in estimating the total area of ice, i.e., ice covering the land of the Antarctica in addition to surrounding sea ice, we should eliminate inside details or shadows that result in black pixels within the area covered by ice.

One straightforward approach to eliminate interior black pixels is to reduce details and shadows in the image using a spatial low-pass filter. Low-pass filtering is a basic image processing technique in which the image is filtered by the means of convolution with a small matrix called the kernel.

Perhaps the simplest two-dimensional low-pass filter is the *averaging filter* defined by an $N \times N$ matrix with all entires equal to $1/N^2$.

- Use MATLAB's "imfilter" function, along with an averaging filter of appropriate size N, in order to eliminate/reduce black pixels inside the white region in each frame.

- Note that if N is too small, the filter will probably leave much of the black pixels uncovered. On the other hand, large N will cause too much smoothing and the edges of the white region might be destroyed (which would negatively affect the accuracy of area estimation). Thus, your choice of N should be a compromise between eliminating interior details and maintaining the edges. You can also apply a filter with small N multiple times.

Q7 What is the size N of the averaging filter you use? (2 pts)
How many times do you apply the filter for each frame? (2 pts)

Fig. 4 Use MATLAB's "imshow" function to display the frame from Task 4 after applying the averaging filter. (5 pts)

Q8 Use MATLAB's "fft2" function to find the two-dimensional discrete Fourier transform (DFT) of your averaging filter. Comment on the result. (5 pts)

Task 6 Estimate the Change in Ice Extent

Fig. 5 Plot the change in ice area compared to its value at the beginning of the duration (i.e., its value on March 21, 2014). The horizontal axis should be the number of days x starting from $x = 1$ (Mar 21, 2014) till $x = 183$ (Sep 19, 2014), while the vertical axis would be the percentage change given by

$$\frac{area(x) - area(1)}{area(1)} \times 100\%,$$

where $area(x)$ is the ice area (in terms of the number of white pixels) at Day x. (10 pts)

Q9 Comment on the change in ice area. (2 pts)

Task 7 Polynomial Regression

- Use an appropriate polynomial function of order $n \geq 1$ that models the change in ice area over time. You can use MATLAB's "polyfit" function.

Fig. 6 Plot the polynomial fit in addition to the data points you obtained in Task 6. (4 pts)

Q10 Write down the polynomial function you use. (4 pts)

Q11 Use your polynomial regression model to predict ice coverage on Sep 30, 2014. (3 pts)

Task 8 Data Collection and Mining

- Download the **Daily Antarctic Sea Ice Extent** dataset provided by the *National Snow and Ice Data Center (NSIDC)* at the University of Colorado, Boulder. You should be able to download the dataset in **CSV** format by following the links at `http://nsidc.org/data/seaice_index/`

Fig. 7 Plot the daily ice extent data for the entire year of 2014, i.e., ice extent in millions of square kilometers versus the number of days (1 – 365). (5 pts)

Q12 State the minimum and maximum ice extent (in millions of square kilometers) throughout the year 2014 along with the corresponding dates. (2 pts) Comment on the change in ice extent with respect to seasons throughout the year. (4 pts)

Fig. 8 Plot the percentage change in ice extent from Mar 21, 2014 till Sep 19, 2014 (as in Task 6 but using the dataset provided by NSIDC). (5 pts)

Q13 Are the results compatible with your calculations from Task 6? If not, explain possible reasons for discrepancy. (5 pts)

Q14 What other techniques could be utilized to extract information from NASA's video in order to obtain more accurate estimate of the ice extent? (4 pts)

3 Image and Video Handling with MATLAB

It is suggested that you use the following MATLAB functions while extracting frames from the video file. The MATLAB help documentation provides sufficient details and examples on the use of each function.

- myVideo = **VideoReader**('videofile.mp4') creates a MATLAB object in order to read the video file "videofile.mp4" using its default frame rate. Note that "VideoReader" creates an object that contains properties of the video file, such as name, duration, and frame rate, but it does not contain the video file itself.
- A = **readFrame**(myVideo) reads the next available frame from the video file associated with the "VideoReader" object "myVideo".
- **imwrite**(A,'imagefile.png') writes the image data A into the file "imagefile.png".
- B = **imread**('imagefile.png') reads image data from the file "imagefile.png".
- **imshow**(B) displays the image data B in a graphics figure.