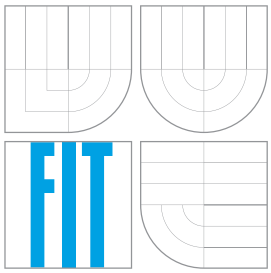
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VISUAL GEOLOCATION OF A STATIONARY CAMERA

VISUAL GEOLOCATION OF A STATIONARY CAMERA

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PAVEL ŠIMURDA

VEDOUCÍ PRÁCE
SUPERVISOR

Doc. Ing. ADAM HEROUT, Ph.D.

BRNO 2014

Abstrakt

Tato práce se zabývá a analyzuje možnosti, kterými je možno zjistit geografickou polohu ze snímků nebo videa pouze za použití vizuální informace z obrazu. Výsledkem práce jsou dvě rozdílné metody geo-lokalizace. První z nich pracuje na principu hledání časů východu a západu Slunce. Hlavní výhodou této metody je její univerzálnost. Funguje s jakoukoliv kamerou umístěnou v externích prostorech a nevyžaduje přítomnost žádných specifických objektů ve scéně. Pro správný výsledek je třeba alespoň celodenní záznam z kamery. Výsledky jsou uspokojivé za každého počasí. Druhá metoda pracuje na základě analýzy stínů ve scéně. Správnou pozici je možno určit, s poměrně velkou přesností, pouze na základě dvou snímků pořízených v různém čase. Tato metoda vyžaduje přítomnost dvou objektů v obraze, které vrhají stín. Přesnosti výsledků navržených metod jsou vyhodnoceny a porovnány. Z výsledků vyplývá, že obě metody lze úspěšně použít pro odhad geografické polohy. Dále byla v rámci práce pořízena rozsáhlá datová sada obrazových sekvencí z volně přístupných webových kamer.

Abstract

In this thesis we address a problem of geo-locating a camera using only visual information of the scene. The aim of the thesis is to study existing methods of geo-location, discuss and show their limitations and suggest possible improvements. We propose two different methods of geo-localization. The first method is based on detection of sunrise and sunset in the image sequence. Its main benefit is that this method can be successfully applied for any outdoor camera and it does not require any specific object in the scene. The second is based on the analysis of shadows in the scene. This method allows estimation of the Sun's position a single image using two objects and their shadows. It is required to have at least two images captured at different time for the correct estimation of the geographical position. We evaluated proposed methods and introduced discussion and possible improvements. Both methods can be successfully used for the geo-location determination. In addition, we created an extensive dataset of time-lapse images from public accessible web cameras.

Klíčová slova

Geolokace, stacionární kamera, klasifikace, den, noc, analýza stínů, detekce stínů

Keywords

Geolocation, stationary camera, day classification, night classification, shadow analysis, shadow detection

Citace

Pavel Šimurda: Visual Geolocation of a Stationary Camera, diplomová práce, Brno, FIT VUT v Brně, 2014

Visual Geolocation of a Stationary Camera

Declaration

I declare that I have written the Master's thesis by myself without undue help from others under supervision of Doc. Ing. Adama Herouta Ph.D. I also declare I have included all literature and publications I have been using as resources for my work.

. . . . . .
Pavel Šimurda
May 28, 2014

Acknowledgement

I would like to express my gratitude and thanks to my supervisor Doc. Ing. Adam Herout Ph.D. who has shown an interest in my work and gave me an important feedback. I would also like to thank to Ing. Markéta Dubská for suggestions in the process of the camera calibration.

Contents

Chapter 1

Introduction

In recent years the rate of public accessible web cameras has grown significantly. These cameras have wide usage in areas of traffic or weather monitoring, security, virtual sightseeing and advertising, to name a few. Thanks to the technology improvement it is easy to mount a camera and provide the real-time transmission to a user in other parts of the world. However, in some applications images from cameras lose their meaning, if the user does not know where the camera is situated e.g. weather or traffic surveillance. Even though public web cameras usually provide its location, in some instances this information is not available. In this case the geographical location (*geo-location*) can be inferred from the imagery itself. This is useful for intelligence agencies, criminal investigations or simply as a proof of concept.

1.1 Motivation

Is it possible to determine the location from video or even from a single image? Figure 1.1 shows a web-based geography game[1] where a user must guess where the image was taken. It might be difficult for a person to determine where was the image taken just by analysing the scene. However, using machine vision and image processing techniques provide possible solutions. Vegetation, climate, signs etc. give guidelines to estimate the location. Although, a more reliable source for geolocation is the position of the Sun. Even thought the Sun is not present in the image, we can calculate the *Sun's elevation* and *azimuth* using the shadows cast by objects in the scene. Consequently, obtaining important clues from the image in order to estimate the geo-location is a difficult computer vision problem. This thesis aims to make a contribution in this field.

Geographical information from a video or an image is important. It can be used for labelling or managing photos. Furthermore the project by RIPS [8] identifies intelligence agencies requirements for determining where a particular video was captured. Most videos and images usually have little to no meta data. Although some cameras are using EXIF [4] information to store settings such as exposure time, focal length etc., the geo-location is rarely provided. It is not always possible to equip camera with GPS technology. Moreover, the process of obtaining GPS signal needs a certain amount of time, which is not practical for a photographer. The idea of using IP based geo-location for web cameras connected to the internet [22] may not be always efficient - especially in large countries with poor network infrastructure.

[1] `http://geoguessr.com/` maps and images provided by `www.google.com`

Figure 1.1: Where was the image taken? Images from public accessible web cameras and the „geoguesser“ game.

The aim of this Master's thesis is to study the methods for estimating the geographical position from video provided by the web camera. Additionally, to find a new method or suggest improvements to existing ones. We chose to improve the geo-location method based on shadow analysis in the scene, and to describe the possibilities of obtaining location from one single image. However, during the experiments a new method of geo-location based on times of sunset and sunrise was proposed.

1.2 Thesis Organisation

- **Chapter 2** describes a general problem of estimating the geo-location. The related theory based on astronomical calculations is briefly explained.
- **Chapter 3** introduces our first method based on finding time of sunrise and sunset as a clue for determining the geo-location.
- **Chapter 4** describes our second method based on the shadow analysis and measurement of the Sun's elevation angle.
- **Chapter 5** gives more details about the astronomical calculations of location based on determined sunrise, sunset and the Sun's elevation angle.
- **Chapter 6** presents the dataset which we used for experiments.
- **Chapter 7** provides details about implementation, used libraries and technologies.
- **Chapter 8** concludes the entire thesis, summarizes proposed methods and discuss possibilities of a future work.

Chapter 2

Camera Geolocalization

The most common guidance for determining the geo-location are positions of celestial objects based on a celestial model. This method of navigation is called astronavigation. The night sky provides us many constellations and stars from which we can determine our position on the Earth. Positions of planets and stars remain the same for certain places at certain time. Nevertheless, for a conventional web camera it is not possible to capture the night sky properly. Therefore, determining location from the positions of night celestial bodies is achievable in a rather narrow range of situations. During days with clear sky the Sun provides a reference point for navigation. An observer can determine the position from the Sun's relative path across the sky.

On the northern hemisphere the Sun goes from east to north and the apparent Sun's trajectory is visible by an observer on the south. For small latitudes the Sun's altitude angle (we understand the Sun's altitude angle and the elevation angle as the same thing) is greater in comparison with high latitudes, where the elevation of the Sun is lower at the same time. In addition, the elevation of the Sun is higher in summer and lower in winter. In the southern hemisphere it is vice versa. Generally the elevation of the Sun is the highest at noon.

The usual procedure in determining the ones location is to measure the angle between two celestial bodies or between the Sun and horizon for the night sky and for day sky, respectively. When the angle is measured, the correspondent position given by date and time of observation is found in the Astronomical Almanac [28]. As we mentioned before, measuring angles in the night sky is out of scope for our situation. It is also not possible to obtain direct Sun observation in every scenario. However, the angle of the sunbeam falling on the Earth's surface could be estimated indirectly from the shadows cast by objects (*shadow-casting object*) in the scene. This is one of the main phenomenon we take in consideration for determining the position.

2.1 Traditional Methods

Travellers and sailors have always had needs to navigate on their journeys. Without prior knowledge of GPS [1] or even spheric geometry, navigation and positioning was achieved by observing the position of celestial bodies. The night sky with moon, planets and stars was the reference point mostly for the travellers. All celestial bodies are following a path across the sky relative to an observer on earth's surface. Paths differ during the year and also in the northern and southern hemisphere. An important fact for our research is that the Sun

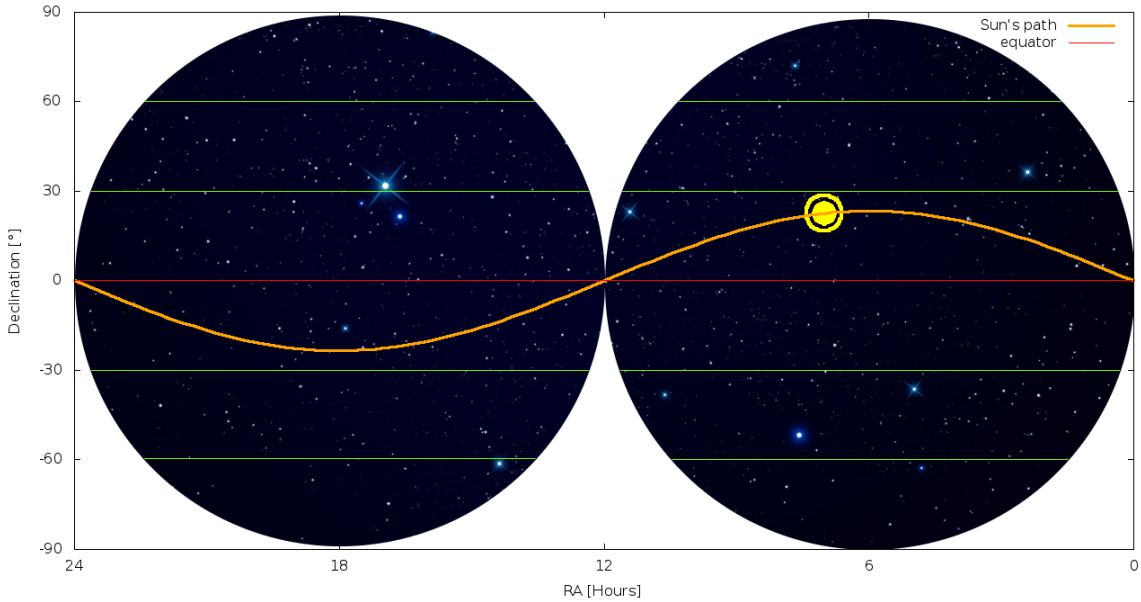


Figure 2.1: Sun's movement along the ecliptic (orange sinusoidal curve). Correspondent position of the Sun on 7th July (yellow point in the diagram).

follows a sinusoidal path relative to an observer (Figure 2.1, the diagram is in equatorial coordinates, which are discussed in the Chapter 5). Surprisingly, ancient Egyptians were very advanced in observing celestial bodies and related geometry and mathematics. The principle of calculation of the Sun's position was known, as we can see in pyramids in Giza. Holes in the walls of pyramids are situated, so the sun shines on the king's grave each year [33].

Many tools for navigation and positioning were invented in the distant past [27]. They were easy to use and very accurate. Many of them are using simple principles of mathematics. With only an accurate watch, a compass, a sextant and the Astronomical Almanac, the position of an observer could be estimated with accuracy close to 0.1 nautical miles which is approximately 185 meters.

Now, we introduce a brief description of some navigational tools. These tools gave initial ideas for determining the geo-location from a machine vision perspective. Similar phenomena are observed within the image scene.

- **Sextant** is an instrument used mostly by seafarers to measure the elevation of the sun above the sea level. But in general it serves for determining the angle between two celestial objects.

- **Sundial** (analemmatic sundial) is an example of basic clock. Using the phenomenon of an object casting a shadow from the rod centred in the surface of a clock. This rod is called *gnomon*. The Sundial must be properly calibrated in order for it to work correctly. In its basic configuration it looses precision due to the fact that Earth's orbit is elliptic, and the apparent Sun's path is sinusoidal (Figure 2.1). Therefore, the time can be either a quarter-hour late or early, depending on the calibration. The correction of time is described by *Equation of the time*. More information about sundials and the Equation of the time can be found in [25].

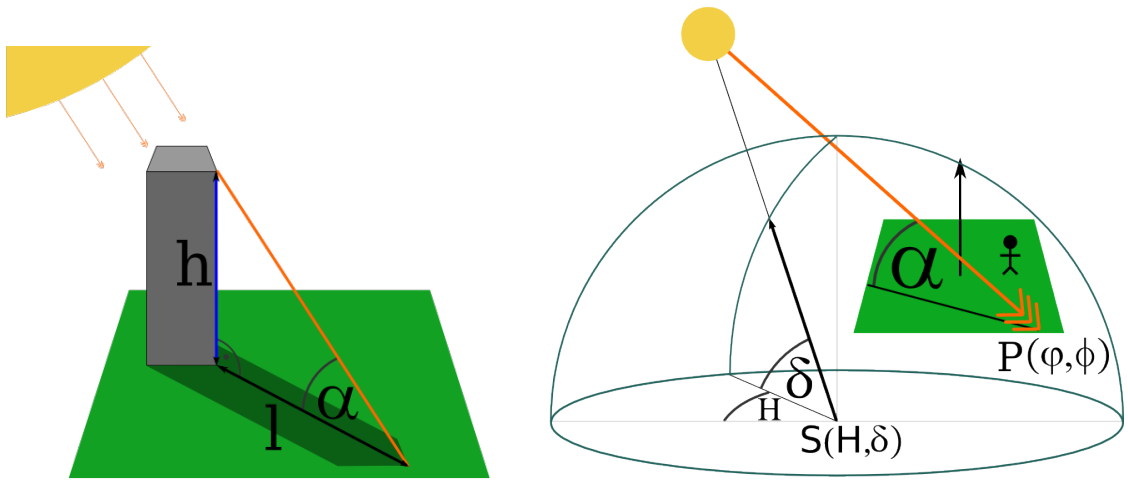
Tools described above rely on the fact that the elevation of the Sun in the course of the day is growing after sunrise and reaches its maxima at midday. After that it is decreasing until the sunset is reached. Moreover, differences in the Sun's path in winter and summer for both hemispheres are taken in consideration.

These traditional navigation principles inspire the modern technologies. Mostly in the world of artificial intelligence or autonomous robot navigation. Another usage of these principles can be found in camera calibration techniques. Cameras are often used as a digital sextant providing the information about its position. The possibility of using Sun's altitude for localization of a robot is discussed in [10]. In fact this approach can be used by the robot on any planet. Astronomical calculations to determine the position from declination angle are easy to adapt for an observer on any celestial body, not necessarily the Earth.

Creation of a digital sundial is an interesting task from the two perspectives. Firstly, it is visualization of the phenomenon in the 3D graphic environment, but this task is more about modelling and 3D rendering. Secondly, it is the process of obtaining the shadow cast by object and gathering of information in order to determine the angle of the sun and horizon. The latter principle is used in proposed method of estimating geo-location.

2.2 Possible Approaches

How is it possible to determine position of a web camera by analysing its video stream? This problem is called *Visual geolocation* - determining the geo-location only with usage of visual information provided by camera. We addressed this issue using two different directions.



(a) (b)

Figure 2.2: **(a)** Sun's elevation and shadow-casting object relationship. **(b)** Position of the Sun seen from the Earth, outline of equatorial coordinate system.

First direction of the approaches is using shadows in the scene. Our initial idea of how to find the geo-location was inspired by a Sundial. With the knowledge of the geo-location (in the traditional geographic coordinate system given by longitude ϕ and latitude φ) and setting the orientation of the gnomon, this device tells the time of the day. That means there is a relationship between the shadow of gnomon and time. Therefore, an inverse method where the time and shadow is known and location is desired can be derived. Shadows in

the scene provide information about the elevation of the Sun (Figure 2.2a).

$$\alpha = \tan^{-1}\left(\frac{h}{l}\right) \tag{2.1}$$

$$\sin(\alpha) = \sin\delta \sin\varphi + \cos\delta \cos\varphi \cos H \tag{2.2}$$

Given an object with a height h and a shadow length l, using Equation 2.1 the elevation angle α is calculated. The relationship between α and the geographical location of the observer (Figure 2.2b) is given by Equation 2.2, where φ is the latitude of the observer, δ is the declination and H is the hour angle. Detailed study of this approach is given in the Chapter 4.

In addition to these simple formulas we consider the earth's surface spherical and the earth's orbit elliptical, which is not entirely true. What we have described here is so called *Simple Spherical Model*.

For more serious equations we must take into consideration the imperfection of the earth's surface. There are many models for geographic calculations. They differ from a specific application and one can not be used as universal. For requirements of this work only models using day length as function of latitude were studied [12, 32], where the CBM model by authors Forsythe et al. [12] gives the most precise numbers. As a matter of fact most of those models are ecological or agronomic researches. Mostly they are used for water, carbon and nitrogen cycles (Forest-BCG model by Running and Coughlan [32]) in the environment, where highly precise representations of gas flow or climate changes is needed.

Another approach proposed in this thesis utilizes the relative Sun's movement as the clue for the location determination. From the video recorded during the course of the day it is possible to find the ratio of length of the day and length of the night. The length of the daytime depends on the observer's longitude. During equinoxes it is about the same all over the world, but in the course of a year it is slightly different for certain latitudes (Figure 2.5, 2.4 and 2.3). Furthermore, times of sunrise and sunset are very important clue for determining the geo-location. Using the Universal Time the exact time of sunrise and sunset is unique for each location on the Earth. However this method is not limited for usage in sunny days only, it does not have the potential for better accuracy than previous method based on shadows measurement. More detailed description of this method is outlined in Chapter 3.

2.3 Existing Approaches

There are two main categories of camera images geo-localization algorithms. First category is based on correlation of the images taken by several cameras with prior known location. Acquisition of the location from EXIF or meta-data falls into this category as well. Second category is more connected with astronomy or meteorology.

- **Reference Imagery**: These methods are based on feature correspondences between multiple cameras. Prior known geographic location of images is required due to this requirement. In addition, some studies are based on a fact, that a dense network of web cameras is available. Therefore, it is likely to have similar objects captured by two different cameras. Some works introduced in this chapter are following the approach proposed by Jacobs et al. [15].

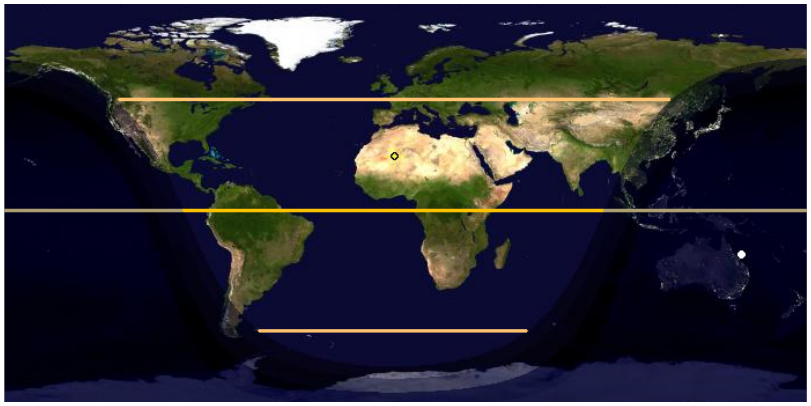


Figure 2.3: The northern solstice - day in the norhtern hemisphere is the longest and night is the shortest.

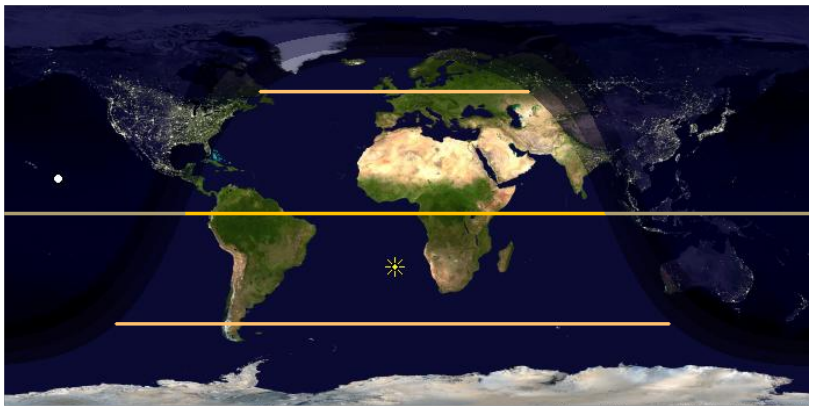


Figure 2.4: The southern solstice - day in the southern hemisphere is the longest of the year and night is the shortest.

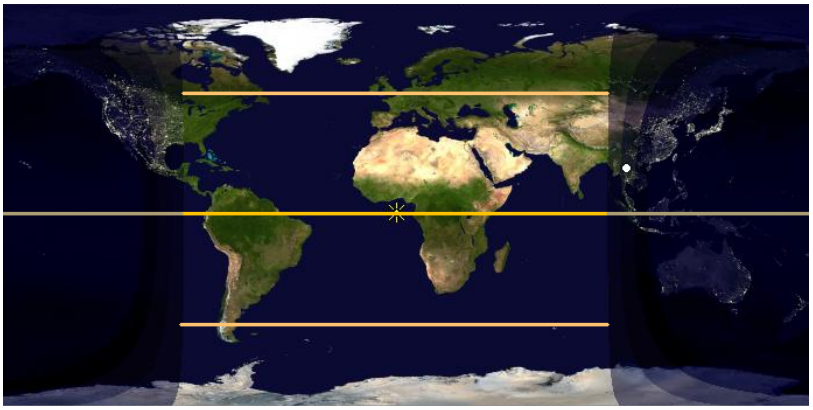


Figure 2.5: Equinox - day and night are about the same length all around the world. Twilight period is longer closer to poles.

- **Sun and Weather**: Some methods require the sun to be visible in order to calculate the sun altitude angle. Other methods use shadow tracking but many of them need manual preparation, such as marking an object casting the shadow. They also require presence of a flat ground plane very often, thus shadow can be tracked easily. Certainly cast shadows provide rich information about the scene and location, yet they are difficult to detect. There are also approaches using meteorological data about weather to find the exact location In this category.

There are several approaches of how to determine the geographical location of web cameras. For geo-locating static cameras Jacobs et. al. [15] use a database collected during a year and apply correlation to the weather forecast maps. In fact they are dealing with a network of cameras (with already known locations) and in addition they propose a creation of a map of weather conditions using this network. Camera position is therefore obtained by corresponding features, matching points or landmarks. More specifically the authors define a synthetic daylight map for the geo-localization. By using the daylight map and geo-registered images from the NASA geostationary Satellite [2] they achieve localization with the mean error of 80km.

The most sophisticated and very precise solution was introduced by Junejo and Forrosh [20]. They are using two-step method, where the first step is the camera auto-calibration. This part is described in authors' previous research [19] of camera auto-calibration techniques. The second step is the geo-temporal localization based on shadows on the ground plane obtained from a calibrated camera. The important part in this work is the camera calibration itself and recovering the vanishing line of the ground plane from shadows. They recover the Image of the Absolute Conic (IAC), which is decomposed into the camera calibration matrix $\mathbf{K}$. Conics are found in shadow trajectories on the ground plane. Using the horizon line and the vertical vanishing point as geometric entities in the image plane, the camera model is recovered. The usual relation between the IAC and the internal parameters of the camera is shown in Formula 2.3:

$$\omega \sim \mathbf{K}^{-T}\mathbf{K}^{-1} \tag{2.3}$$

They present a new relation that is more intrinsic to the IAC. They only need the camera calibration matrix $\mathbf{K}$ to measure the angle between two projection rays. Knowing ω is almost equivalent of $\mathbf{K}$ (More information about the calibration matrix is in Section 4.3). They come up with several solutions for the recovery of the vanishing line:

- The shadow-casting object is visible
- The shadow-casting object is partially visible
- The shadow-casting object is not visible

The shadow detection itself is not present in their research and shadows are selected semi-automatically. Nevertheless, a robust estimation of vanishing line in cases where shadow cast on the ground plane might not be easy to localize is presented as well. Very precise estimation of the latitude from the azimuth and the altitude angle of the Sun is proposed. Determination of the longitude is based on a temporal correlation with three shadow points. The proposed method is based on the robust camera-auto-calibration [19] which leads to precise evaluation and calculation of shadow lengths and angles in the camera scenes. Final

results are very precise: for the worst case scenarios they achieved an average error rate of less than 1% with respect to real coordinates. The average error in latitude is 1.53° and average error in measurement of the declination angle is 1.72°.

Another approach for web cameras geo-localization by Jacobs et al. [17] examines relation between the Sun's elevation angle and the light intensity of a scene. This paper aims for evaluation of the position of the camera based on a long-term time-lapse imagery. As the Sun moves, the intensity of a scene changes. When the Sun is rising (zenith angle increases) the intensity of the image is increasing as well and vice versa. In general, they are achieving the determination of the geo-location by calculating it from the intensity sequence with time stamps. They are achieving noticeable results by parametrizing the relationship between zenith and intensity and fitting it in their model.

Recently presented approach by Jacobs et al. [16] debates potential of the cloud motion captured by web cameras as a calibration cue. Showing the potential to obtain camera position by finding corresponding images with weather forecast satellite data.

Chapter 3

Day Night Detection

This chapter gives a detailed description of the strategy for determining the geo-location which is based on the length of the day and times of sunrise and sunset. This approach is easy to use in situations where we have access to video or time-lapse image series from a camera. Although it does not seem as a very robust method, it gives precise estimation of the position regardless of the weather or presence of the shadows in the scene.

Initial ideas of how the method was conduced is outlined in Section 3.1. The method itself begins with feature extraction for classification of images into day and night classes (Section 3.2). For some images a method for sky region segmentation is used (Section 3.3). The final Section 3.4 deals with problems and evaluates precision of the results.

3.1 Method Outline

As mentioned before, the determination of geographical coordinates could be divided into two main tasks which are naturally calculating the latitude and the longitude. We can get several important moments such as the time of sunrise and sunset, twilight, length of the *night time* and length of the *day time* from the time-lapse sequence obtained within one day. We can approximate the position on the globe from the length of the day time in a specific day. This information gives us a clue whether the place is in the southern or northern hemisphere. Moreover given accurate set of sunset and sunrise times unambiguously identifies the geo-location of the web camera (excluding areas near poles, where the Sun never sets in certain seasons.).

After the daybreak happens, a gradual increase in brightness is proceeded until the point when the brightness stabilize into typical day time scene. The situation is reversed before the sunset (Figure 3.1). Both transitions take about the same time. Depending on the location and date the transition time differs. These transition is called twilight and it is known as dawn and dusk, transition from night to day and from day to night, respectively. More about these phenomena is written in Section 5.3. The length of a twilight is an important clue, because during equinoxes the day time is approximately of the same length everywhere in the world(Figure 2.5), but further from equator the place is, the twilight is longer.

This method requires a time-stamp of when the image was taken for determining the location of a stationary camera. In fact, only images from periods around twilights are sufficient for the correct geo-localization. This method of geo-localization allows to use an automatic approach. Hence an algorithm for the detection of the night time and day time is required. We can obtain a sequence of consecutive time-stamped images, so called time-

Figure 3.1: Sunset (dusk) is typical for a gradual decrease in brightness in the image.

lapse from the the camera. Dataset provided from the web cameras is more described in Chapter 6.

Proposed method of the geo-localization needs to determine whether the image scenery is in a night or day time. Thus a basic binary classifier was developed. The general method presented in this thesis works with a single image only. Experimenting with this classification was done using two different machine learning principles. The one we adopted uses Artificial Neural Networks (ANN Section 3.2). The result of the classification is a degree of membership into day and night class (for example a completely black image will most probably result into: $(day, night) = (0.00, 1.00)$). The Support Vector Machines algorithm uses a non-probabilistic binary linear classifier. With usage of SVM algorithm we did not achieve a better precision, even though problem looks like a basic linear segmentation or classification task. When SVM was using same features the results were inefficient.

The method itself is not only a day/night classifier. It is also necessary to work within a *context* of one day and find the dependencies of the series of images and omit errors caused by the classification. Working with the context brings benefits into the solution. We understand the logical time order, e.g. it is impossible that algorithm evaluates the change between night and day four times in the course of one day, or the sunset can not precede the sunrise etc. In ordinary circumstances it is difficult to classify image which is darkened by cause of bad weather, shadow, clouds etc. With usage of the context it is not a problem to eliminate false responses from the classifier. Twilight is detected in the period for which images have uncertain classifier response. More about the twilight definition from the astronomical point of view is discussed in 5.3. The sunrise occurs in the context when the scene turns from dark into brighter colours and vice versa for the sunset.

We do not call the 24 hour cycle itself as *the Length of daytime*. In astronomical terms it is a period during the day which starts when the upper rim of the Sun appears at the horizon and ends by the moment when it disappears on the opposite side. It can be calculated from the CBM model described by authors in [12]. Ordinary models are not very precise because of the fact that our planet is not easy to describe as a geometric body. There are many deviations which are difficult to follow in the equations and calculations. Therefore many of approximations in the description of the Earth's surface were done as can be seen in Equations 3.1.

$$\theta = 0.216318 + 2\tan^{-1}(0.9671396\tan(0.00860 * (J - 186)))$$
$$\phi = \sin^{-1}(0.39795\cos\theta)$$
$$D = 24 - \frac{24}{\pi}\cos^{-1}\left(\frac{\sin\frac{p\pi}{180} + \sin\frac{L\pi}{180}\sin\phi}{\cos\frac{L\pi}{180}\cos\phi}\right) \tag{3.1}$$

Authors claim that this formula is a precise function of day length time based on the latitude and given day of the year, where D is the day length, L is the latitude and J is the day of the year. Given constants are taken from the CBM model in order to eliminate imperfections of the earth surface. Hence we can obtain the day length and the day of the year. This formula gives only the latitude as result. It does not take into consideration the usage of time-stamped images with the Universal time. Moreover the constant correction for elliptical trajectory of the Sun causes inaccuracy. Therefore, different and more convenient model was adopted (Section 5.1).

Existing Approaches

Naive approaches of the day and night classification are often based on the thresholding of average intensity values in the image. This is not a very precise method and it needs many corrections and improvements. There are not many approaches dealing with the day/night classification. Mostly they are connected to indoor-outdoor image classification problems. Many of them deal with day/night classification along with querying images by contest as authors of [36] recall. For them day and night classification is not the only output. Method developed by Soo Kim et al. [23] is only based on one feature - CIELUV histogram showing the deviation in luminance channel. Day images are much brighter than the night ones and that is the main classification criterion. Wang and Hu [37] choose chroma values from the HSV image histogram as features. This method is also able to classify sunrise and sunset. They use training set containing more than 11000 images. Other methods rely on astronomical phenomenon of changes in the sky colour during the day. Hseih and Chen [14] use sky region as the main feature for the recognition of day time or night time image.

3.2 Classification Features

An accurate determination of sunrise and sunset from already classified images into day/night category is not a difficult problem. Unlike the classification itself. We need robust features which indisputably categorize images for a precise classification. From the exhibition of the scene it is noticeable that daytime images tend to be in brighter colours. On the other hand the night time images are darker. The most important feature is based on average image intensity. We are using both the *average image intensity* and the *mean intensity value*. Unfortunately, these two features are not enough for a precise classification.

This method mainly relies on the histogram analysis. In Figure 3.2 we can see a typical day time image with its *RGB* and *HSV* (Hue is shown in the purple colour, value in gray and saturation in orange.) histogram. The distribution of values in the histogram is in the middle of the graph. In the night time image (Figure 3.3 the distribution is almost reversed. In other words, during the day time, the Sun enlightens the scene and colours are bright with more value and saturation, and that is why the HSV histogram was chosen. On the contrary, the distribution of colours in the night time image contains value and saturation. There

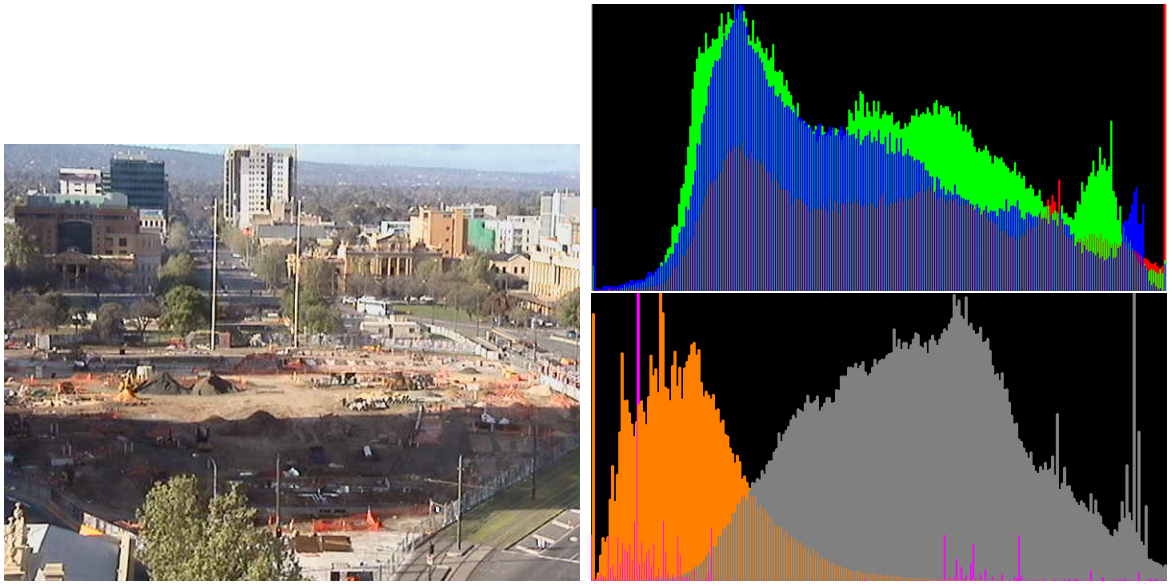


Figure 3.2: Day time image with characteristic RGB and HSV histogram values distribution

are images where distribution of histogram does not fit the observed facts. Therefore, in the feature vector we use raw values from histogram of 16 bins (image histogram is usually 256 bins each width of 1). The histogram feature sub-vector for single channel image is described as:

$$Hist_{ch} = \{bin_0, bin_1 \dots bin_{15}\} \tag{3.2}$$

where bin_n means number of pixels in the range of $[16n, 16n + 15]$ for given channel, and $Hist_{ch}$ is the feature we use in the feature vector. Note that for the RGB (HSV) three channel histogram we will have three of these feature sub-vectors.

Another important finding is that night time images from urban areas, where web cameras are often situated, usually contain street lights. These lights are causing an artificial enlightenment of the scene with the spectrum different from the spectrum emitted by the Sun. Lights in the night scene create bright blobs in the image, rest of the image is usually dark. Therefore, other features are introduced. The First one is *number of white blobs*. Second on is giving a ratio of bright and dark (more precisely close to white and close to black) pixels, where threshold values were found empirically:

$$ratio_{bw} = \frac{I_W}{I_B} \tag{3.3}$$

$$I_W = \begin{cases} 0 \texttt{ if } I_{x,y} < 220 \\ 1 \texttt{ otherwise} \end{cases} \tag{3.4}$$

$$I_B = \begin{cases} 0 \texttt{ if } I_{x,y} > 35 \\ 1 \texttt{ otherwise} \end{cases} \tag{3.5}$$

Next feature is based on the astronomical phenomenon of the colour of the sky. Seafarers on the open sea use the horizon as a reference point. After the *nautical twilight* the colour

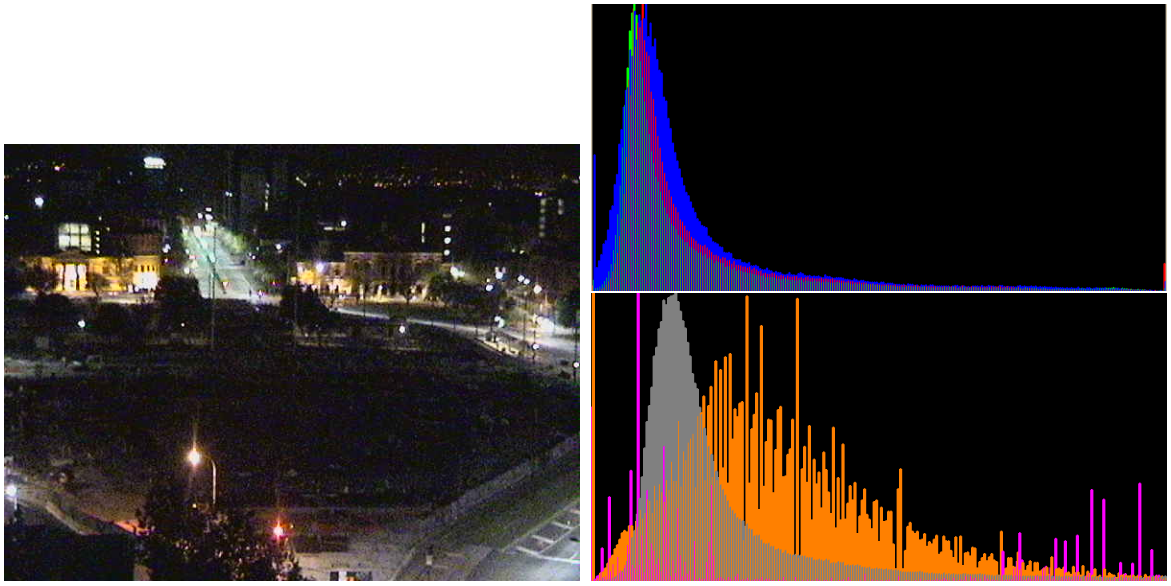


Figure 3.3: Night time image with characteristic RGB and HSV histogram values distribution

of the sky is too dark and it is no longer possible to recognize where the horizon is. We use a simple method for the sky segmentation in order to find an average colour of the sky and intensity ratio between sky and the rest of the image. The colour of the sky is the bluest in the noon. Closer to morning or evening colour is getting darker. During the twilight the colour of the sky tends to change from dark blue through orange into red. This is caused by scattering of the sunlight or the starlight. We introduce *Sky not-Sky ratio* SnS_r:

$$SnS_r = \frac{I_{sky}}{I_{ground}} \tag{3.6}$$

where first the sky segmentation algorithm is applied and it split the image into two segments. The sky-segment with x_{sky}, y_{sky} pixels and the ground-segment. These two segments are disjoint. Naturally clouds in the scene cause problems.

$$I_{sky} = \sum I_{x_{sky},y_{sky}} \tag{3.7}$$

$$I_{ground} = \sum I_{x,y} \tag{3.8}$$

In scenarios where the sky segmentation fails or where the sky is not present in the image the simple assumption that ground is located at the bottom part of the image and the sky is located in the upper part of the image is made. The top quarter of the image is chosen as the sky area.

Finally we introduce the feature vector for the classification method with histogram bins, ratios, average sky region colour, average intensity and mean intensity:

$$\mathbf{F} = \{Hist_{rgb}, Hist_{hsv}, blobs, ratio_{bw}, SnS_r, Sky_{avg}, I_{avg}, I_{mean}\} \tag{3.9}$$

Two basic cases need to be considered for detection of daytime scenes. First when the sky and potentially the sun path is visible. In this case we can use proposed sky segmentation

technique (Section 3.3) as a part of the feature extractor. Second when the sky is not visible in the image scene. This case is more general and we can use only the information about how „bright“ the scene is.

The machine learning algorithm uses training set of 350 day images and 350 night images from the created dataset (Chapter 6). Sky is not always visible in those images. The entire classification process is as follows. First the sky area within image sample is detected. Then the feature vector is obtained and finally the backpropagation algorithm for training the classifier is performed. As algorithm for the machine learning and classification we selected ANN, because of its robustness and available implementation in the OpenCV library.

Artificial Neural Networks

Artificial neural networks is a computational model suitable for machine learning and pattern recognition [5]. Due to its capability of accepting multidimensional input it was chosen as a method for recognising day and night images. It is based on artificial neurons which are derived from real biological neurons. Artificial neurons are interconnected in several layers, where each neuron has n inputs but only one output.

Commonly used layout is where there are two layers of neurons, where neurons in the first (hidden) layer are connected to inputs (e.g pixels of the image). By applying weights and transforming by activation functions these inputs proceed into the second (output) layer of neurons. Number of neurons in the second layer is given by number of classes. After the application of activation functions in the second layer, outputs of the network are obtained. As an output the vector of responses is given. Each element represents the predicted response for a sample to belonging into specific class.

3.3 Sky Segmentation

The sky segmentation is an important part of the feature extraction method. The sky colour is changing during the day and it is a very significant feature for the daytime recognition. For the set of images from the same static camera it is relatively easy to find a *sky region*. It can be calculated once and used for a long term period. Consequently, it is kept as one of the camera's parameters leading to scene segmentation. The algorithm of segmentation simply converts the image taken by camera into HSV colour space, because it is easier to filter specific colour values. Pixels which are not close to the blue colour of the sky are omitted and everything that remains is marked as a *candidate sky region*. The sky region is usually located on the top of the scene, that is why we consider only upper half of the image for the sky detection. It is not very likely that small separated sky region appears on the bottom of the image. We use modified *Floodfill* algorithm with adaptive threshold for determining the candidate sky regions. Once the colour of thy sky is found, the sky region is extended using our Floodfill method. A simple improvement of adding the possible sky colours dynamically was considered, but it did not lead to better results.

$$SkyBlue_{(x,y)} = \begin{cases} 1 & \texttt{if } \ src_{(x,y)} = \texttt{inRange}(HSV_{BlueMin}, HSV_{BlueMax})) \\ 0 & \texttt{otherwise} \end{cases} \tag{3.10}$$

The algorithm is an iterative method. It works for a single image as well as for the image sequence. The more images from the sequence algorithm process, the more precision

is reached. First it finds the candidate sky regions and selects only consistent large regions of *SkyBlue* colour. Basically it is filtering a colour from the HSV colour space expressed as in the Equation 3.10. Where ranges for HSV channels were derived experimentally as follows:

$$\begin{aligned} Hue &: (Hue_{min}, Hue_{max}) \\ Saturation &: (Hue_{min} - 80, Hue_{max} - 100) \\ Value &: (90, 113) \end{aligned} \tag{3.11}$$

$$\begin{aligned} Hue_{min} &\in \{145, 190, 235\} \\ Hue_{max} &\in \{175, 225, 248\} \end{aligned}$$

It is important to mention that Hue and Saturation values are dependent in our ranges. HSV values follow the scheme of implementation in the OpenCV, therefore it might not be compatible with different representation of HSV models.

If selected regions are overlapping with the candidate sky regions from the previous image, they are unified until the last image is reached. That means the sky region will grow until its completely detected (It can also grow into the bottom part of the image). In urban areas buildings, signpost, banners and many other objects can be demolished, moved or changed. That is why a verification of the segmentation should be performed from time to time, even if the sky is already found and the scene is segmented.

(a) (b)

Figure 3.4: Results of the sky segmentation algorithm. **(a)** An example of correctly detected sky region from the image sequence. **(b)** Areas marked with red colour should be detected but algorithm fails.

Sky Segmentation Problems and Results

The main problem is that in the real scenarios the sky region is not necessary continuous. It might be covered by vegetation or variety of objects. Naturally the algorithm starts in the top left corner of the image. Usually the sky is touching the upper boundary of the image and the segmentation takes the best effect there. On the other hand this is not a strict rule.

In Figure 3.4 we can see results of the sky segmentation algorithm for one day sequence in two different places. In Figure 3.4a the sky region is correctly recognised. Clouds here do not cause fails because the sequence of the images is long enough and each part of the sky is uncovered during the sequence at least once during the day. Unlike the second case (Figure 3.4b), where the algorithm fails. Red area was not marked as the sky region but it should have been. This happened because that part of the sky was constantly enlightened either by the Sun or the artificial light during the day and it never turned into the blue colour. Applying the algorithm on another day will most probably lead to correction of this failure.

Overall this algorithm can not be applied generally for any camera, because thy sky may not be present in the scene. Moreover, a simple division of the image into top quarter where the sky is situated is sufficient for the classifier.

3.4 Experimental Results and Evaluation

The main problems in the classifier of day and night time images arise in the time of dawn and dusk. The classifier is not strong enough to overcome this problem. Sometimes it is a very complicated task even for a human, as can be seen in Figure 3.5. Experiments with classifier including the *twilight class* were done. Unfortunately this brought only more failures into classification.

Figure 3.5: Sequence of the images in the morning during sunrise. Night vision mode of the camera (switching into grayscale image in night) and bad weather causes the classification error. It is difficult to determine the precise moment of sunrise from this sequence.

Web cameras are usually of poor quality. Their reactions to the light conditions are very different. When the camera is enlightened, usually by the Sun or an artificial light, the exposure changes and the image is extensively enlightened (Figure 3.8b). This happens also when sunrise occurs and the scene switches from night time into brighter day time. Some web cameras have a sensor which changes the image type from the regular *RGB* into *Grayscale*. Most of the cameras react to the street light, by changing focus or exposure

(Figure 3.6a). There is a noticeable step change in the colour of the scene in the time when street lights were switched off. In the case represented by Figure 3.6a it is very hard to determine whether the image represents night or day. A person would say that it is day based on the fact that the lights are on and there is noticeably dark background. In fact if lights were turned off, the image would be almost identical to Figure 3.6b.

(a) 7:36 (b) 7:38

Figure 3.6: Problems with exposure, street lights and colours in short period of time - step change in the feature vector. **(a)** Night image at 7:36. **(b)** Day image at 7:38.

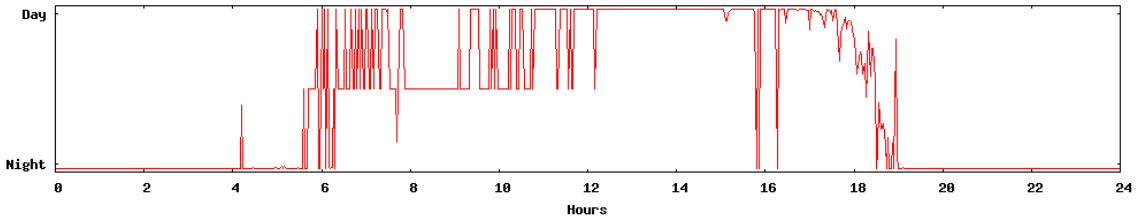


Figure 3.7: Response of the day/nigh classifier for one day.

Another classification problem is caused by sudden changes of weather. During a sunny day with a noticeable change of illumination in the scene (caused by clouds or a storm), classifier evaluates image incorrectly. Nevertheless these problems are detected as peaks in the graph and corrected easily. Typical and unwanted detection error in the sequences is when the day begins with a bad weather. It happens during the dusk, when clouds may cause darker scene as it is shown in Fig 3.5. Consequently night might be prolonged by this error, which may lead to serious mistakes in the calculation of the position. In this case the night vision mode is causing an error in the detection as well. Large enlightened areas in the night time images cause false response of the day time image. This error occurs because a similar image is not present in the training set. Extending the training set with more images of certain scenarios will fix this issue.

(a) (b)

Figure 3.8: Problems in the day/night classification. **(a)** Due to the cloudy or bad weather the classification is not always precise. **(b)** During sunset the image scene is enlightened which causes an error.

In Figure 3.9 we can see responses of the classifier for the set of images within the one day. Images are taken every 2 minutes, x-axis denotes time of the image. This is an example of a graph, where several classification errors occur until midday. Before the dawn around 16 o'clock there are two errors as well. Those are caused by the Sun's enlightenment. Errors in the beginning of the day are caused by the bad weather.

Figure 3.9: False day response caused by insufficient training set.

Evaluating the correctness of the proposed method is a difficult task. The error we calculate is a difference between detected sunrise or sunset time ($Time_{rise}$ or $Time_{set}$) where the mean sunset or sunrise time given by astronomical calculations (Chapter 5). Sunrise is not a one time event, it usually lasts around 4 minutes depending on the season. Twilight is not a single event as well but it is a period of time. Therefore, this evaluation is not very informative. Due to different types of twilights it is important, that the method finds the $Time_{rise}$ and $Time_{set}$ during the twilight period (it can last up to one hour depending on the season). Example of the lengths and varieties of twilight is shown in Figure 3.16[1]. It shows how long each type of twilight last. The output of the method is a time range

[1] Image is taken from the www.suncalc.net

during which the sunrise occurs $Time_{rise} \in (rise_{start}, rise_{end})$. As an example we show a case study performed on three different cameras. Results are displayed in Table 3.1, 3.2 and 3.3.

00:00—02:05 — night
02:05—03:26 — astronomical twilight
03:26—04:21 — nautical twilight
04:21—05:02 — civil twilight
05:02—05:06 — sunrise
05:06—20:39 — daylight
20:39—20:43 — sunset
20:43—21:23 — civil twilight
21:23—22:19 — nautical twilight
22:19—23:40 — astronomical twilight
23:40—00:00 — night

Figure 3.10: Day is divided into several astronomical periods. Important are types of twilights and times of occurrences.

Experimental evaluations revealed that the $Time_{rise}$ and $Time_{set}$ detected by our classification method correspond the mostly to the beginning of *civil twilight* in the morning and to the end of civil twilight in the evening. The mean error of $Time_{raise}$ in comparsion to the end of civil twilight is 8 minutes. For the $Time_{set}$ the error is 10 minutes. The mean error of the day length is therefore 18 minutes. In extreme cases, where twilight lasts for very long time the error can be up to 35 minutes. We asked three people to label sunrise and sunset in the set of images. Their mean error in labelling $Time_{raise}$ and $Time_{set}$ was 7 minutes. Our method seems to be precise in comparison with the human response.

Figure 3.11: The first case study. The camera is located in Františkovy Lázně - Czech Republic

We introduce a case study from three different web cameras. The case study is the most important output showing how the method based on the day/night classification works in the real scenarios. We briefly describe each case. The the results are shown in tables below. In each table φ_M and ϕ_M show the estimated geo-location after the analysis of image sequences from three different days. M_{dist} is a distance between the physical location of the camera and the location estimated by our method. Each table shows the mean error for longitude and latitude. Real times of sunrise and sunset (more precisely nautical twilight, civil twilight and official twilight) are present in the table for each day. The beginning and the end of the sunrise period (same for the sunset periods) detected by our method

are present in the 4th column for the each date. The highlighted value is used for the calculation of the final geo-location. The algorithm does not detect the twilight period in every scenario. The partial results of the estimation for the each date are presented in the last column.

The first camera (Figure 3.14) is situated in the Czech Republic. We used three days (26th March, 31st March and 7th May) for the geo-location determination. The final result is very precise, the estimated location is only 91.11 km distant from the physical place of the camera. The mean error in longitude is 0.7530 °and the mean error in latitude is 0.7673 °. More detailed results and partial results for each day are in Table 3.1. In the map in Figure 3.12 we show the estimated area, where the red marker is the physical location of the camera, the green marker is result provided by our method. The highlighted area shows three estimations calculated from the three days.

Location	Date	Real sunrise sunset	Detected sunrise sunset	Determined coordinates	Distance Error
Františkovy Lázně	26th	5:29 - 18:33	5:38 - 18:28		182.83 km
$\varphi = 50.1173\,^\circ$	Mar	6:01 - 19:05	6:18 - 18:28	$\varphi = 48.55\,^\circ$	$\varphi_e = 1.5673\,^\circ$
$\phi = 12.3509\,^\circ$	2014	6:05 - 19:44	**5:58 - 18:28**	$\phi = 13.11\,^\circ$	$\phi_e = 0.7591\,^\circ$
$\varphi_M = 49.35\,^\circ$	31st	6:20 - 19:39	6:10 - 19:50		83.31 km
$\phi_M = 12.87\,^\circ$	Mar	6:53 - 20:12	6:46 - 20:48	$\varphi = 50.00\,^\circ$	$\varphi_e = 0.1173\,^\circ$
$M_{dist} = 93.11$ km	2014	6:05 - 19:44	**6:32 - 20:08**	$\phi = 13.50\,^\circ$	$\phi_e = 1.1491\,^\circ$
Mean:	7th	5:01 - 20:37	5:08 - 20:38		73.16 km
$\varphi_e = 0.7673\,^\circ$	May	5:39 - 21:15	6:38 - 20:40	$\varphi = 49.5\,^\circ$	$\varphi_e = 0.6173\,^\circ$
$\phi_e = 0.7530\,^\circ$	2014	6:05 - 19:44	**5:48 - 20:39**	$\phi = 12.00\,^\circ$	$\phi_e = 0.3509\,^\circ$

Table 3.1: Partial results for the first camera situated in the Czech Republic.

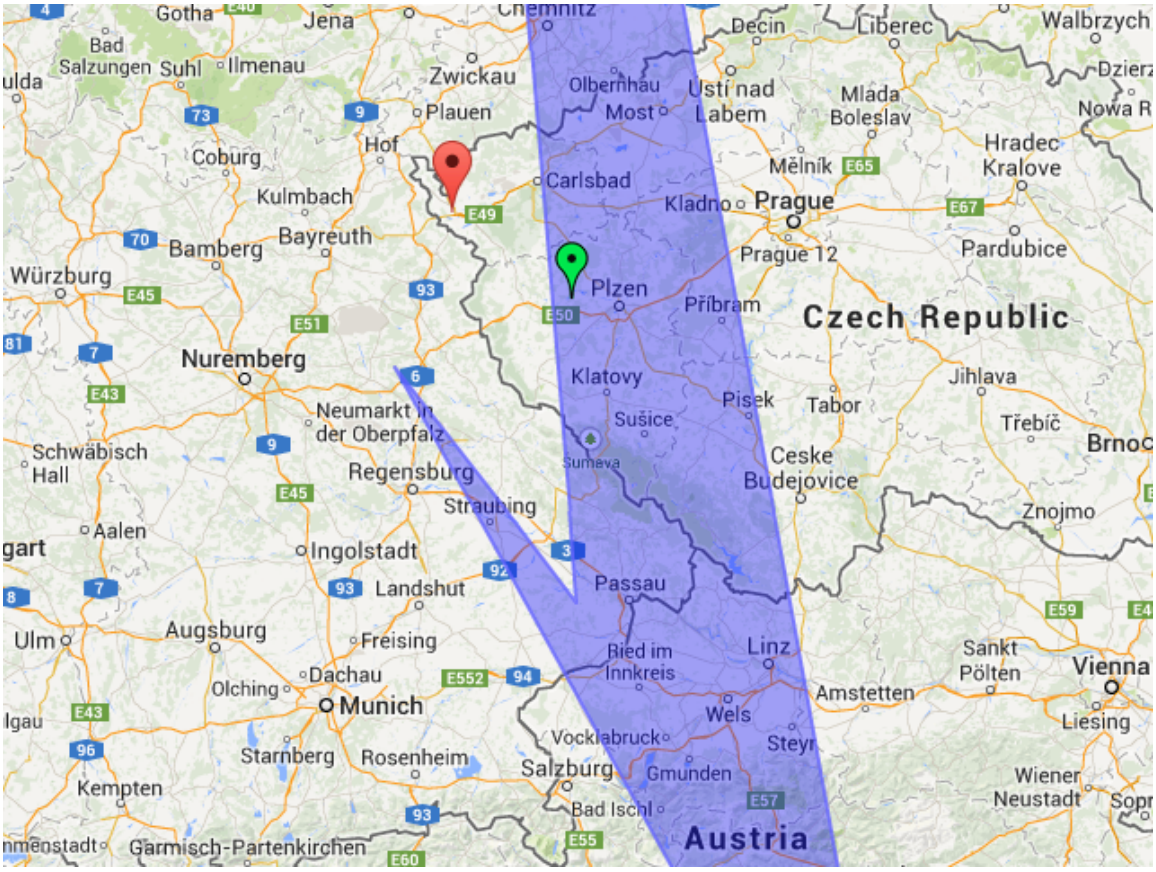


Figure 3.12: The first camera estimation. The red marker is real position of the camera. The green marker and blue are are estimations made by our algorithm

The second camera is located in the island Gran Canaria. The estimation of the location is not very precise compared to the first case. It is caused by the problems in the classifier. The responses of the classifier are influenced by the cameras *nigth time mode* and problems with exposure, as was discussed in Section 3.4. This case is an example where the proposed method of geo-localization gives quite inaccurate results.

Figure 3.13: The second case study. The camera is located in Gran Canaria - Las Palmas, Spain

Location	Date	Real sunrise sunset	Detected sunrise sunset	Determined coordinates	Distance Error
Gran Canaria	2nd	8:26 - 19:43	**8:24 - 9:02**		976.10 km
$\varphi = 28.0526\,^\circ$	Feb	8:50 - 20:07		$\varphi = 33.0\,^\circ$	$\varphi_e = 4.9474\,^\circ$
$\phi = -15.5566\,^\circ$	2014	8:53 - 20:35		$\phi = -7.25\,^\circ$	$\phi_e = 8.3066\,^\circ$
$\varphi_M = 34.05\,^\circ$	7th	8:24 - 21:27	**8:24 - 21:04**		964.51 km
$\phi_M = -15.30\,^\circ$	Apr	8:48 - 21:47		$\varphi = 36.50\,^\circ$	$\varphi_e = 8.4474\,^\circ$
$M_{dist} = 1087.59$ km	2014	8:50 - 22:14		$\phi = -18.00\,^\circ$	$\phi_e = 2.4434\,^\circ$
Mean:	18th	8:12 - 22:29	**8:00 - 21:20**		708.03 km
$\varphi_e = 6.0007\,^\circ$	Apr	8:36 - 22:53		$\varphi = 32.66\,^\circ$	$\varphi_e = 4.6074\,^\circ$
$\phi_e = 5.2845\,^\circ$	2014	8:38 - 23:21		$\phi = -20.66\,^\circ$	$\phi_e = 5.1034\,^\circ$

Table 3.2: Partial results for the second camera situated in Gran Canaria

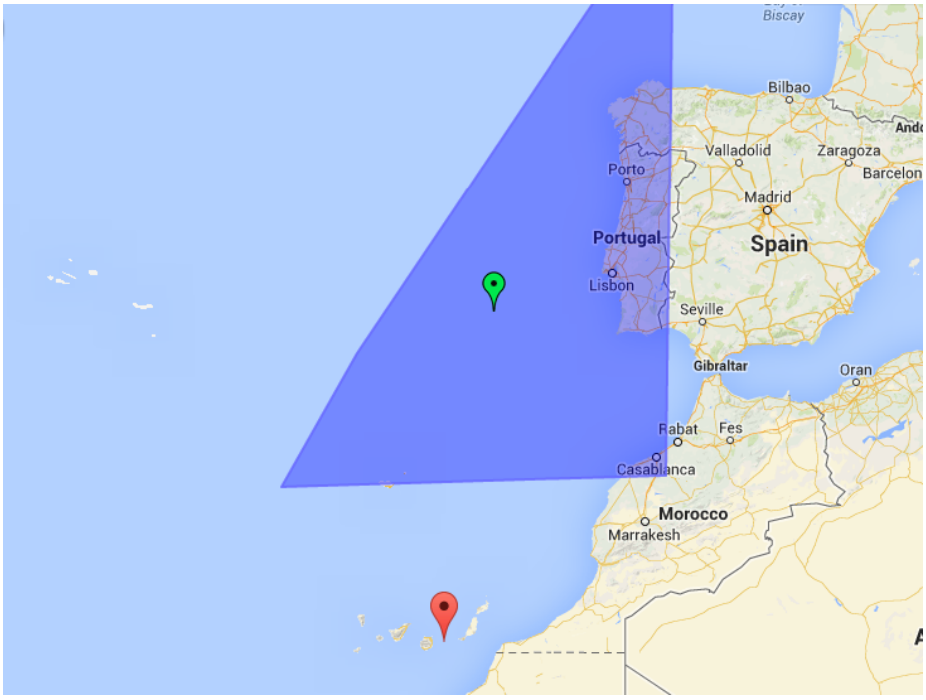


Figure 3.14: The second camera (Canary Islands) estimation. The red marker is real position of the camera. The green marker and blue are are estimations made by our algorithm

The inaccuracy in the third case is caused by failures of camera. The error happened because of the camera's blackouts during a day. We show the third case as an example of error which is not caused by classifier but by an inevitable accident.

Figure 3.15: The third case study. The camera is located in Adelaide - Australia

Location	Date	Real sunrise sunset	Detected sunrise sunset	Determined coordinates	Distance Error
Adelaide	1st	20:06 - 11:03	**20:52 - 11:02**		998.63 km
$\varphi = -34.9261\,^\circ$	Jan	20:36 - 11:33		$\varphi = -26.00\,^\circ$	$\varphi_e = 8.2961\,^\circ$
$\phi = 138.6000\,^\circ$	2014	20:39 - 12:09		$\phi = 140.00\,^\circ$	$\phi_e = 1.4000\,^\circ$
$\varphi_M = -32.44\,^\circ$	7th	22:38 - 10:34	**22:44 - 10:38**		517.96 km
$\phi_M = 138.43\,^\circ$	Apr	23:04 - 10:59		$\varphi = -37.33\,^\circ$	$\varphi_e = 2.4039\,^\circ$
$M_{dist} = 276.21$ km	2014	23:06 - 11:28		$\phi = 143.55\,^\circ$	$\phi_e = 4.9500\,^\circ$
Mean:	19th	22:47 - 10:18	22:56 - 10:28		307.10 km
$\varphi_e = 3.8754\,^\circ$	Apr	23:14 - 10:44	23:02 - 10:36	$\varphi = -34.00\,^\circ$	$\varphi_e = 0.9261\,^\circ$
$\phi_e = 3.1667\,^\circ$	2014	23:15 - 11:14	**22:59 - 10:32**	$\phi = 141.75\,^\circ$	$\phi_e = 3.1500\,^\circ$

Table 3.3: Results of the Sunrise/Sunset method of geo-localization: The first case.

3.5 Discussion and Conclusions

Many of the problems in day/night classification algorithm are caused either by camera properties or inaccuracy during dawn and dusk. The solution can be changing the feature vector and adding more significant features for images taken during twilight. On the other hand the fail-rate of the classification method is under 3%. But average error in estimation of sunrise or sunset time is 9 minutes. A possible improvement for the classification method can be more robust evaluation of the twilight period. This seems to be a challenging problem. From experiments we did, the conclusion is that even for an expert, who is knowledgeable in this field, it is difficult to choose the correct time of sunrise due to lack of visual clues or poor quality of the scene.

We showed how the method works on three different cameras. In the worst case scenario the estimated distance is less than 1000 km distant from the physical place of the camera. The method is not suitable for the precise geo-location. However it provides an estimation of a country, where is the camera located. This method does not require the shadow-casting

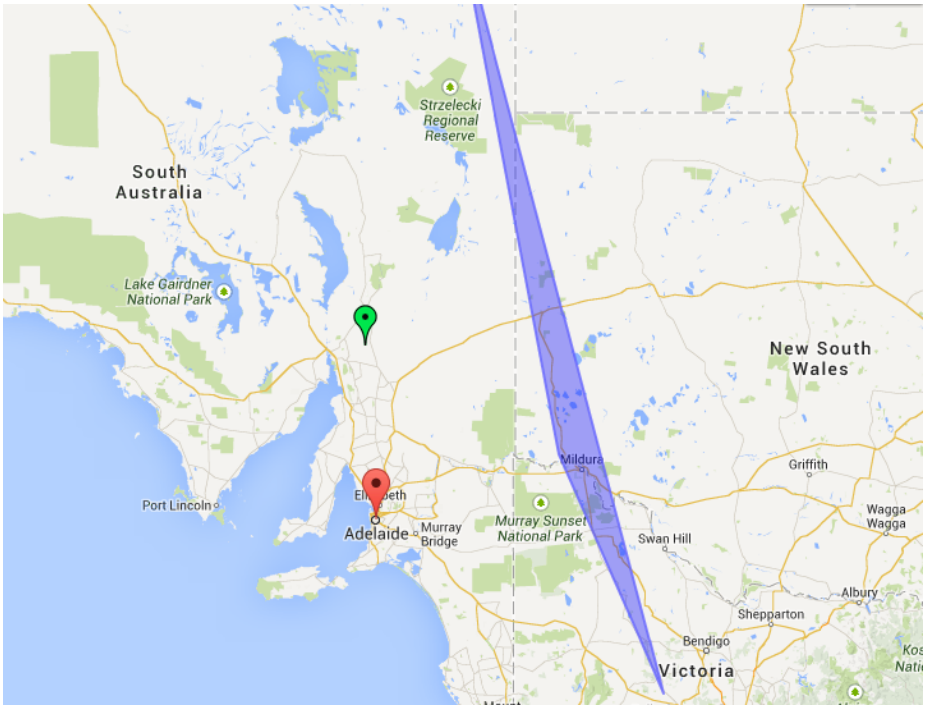


Figure 3.16: The third camera (Adalaide - Australia) estimation. The red marker is real position of the camera. The green marker and blue are are the estimations.

object in the scene which is the main advantage. In other words, it can be used for any camera regardless the condition of the scene.

In the best case scenario it is possible to determine the geo-location with deviance of less than 100 km from the physical place of the camera.

The deviance from the real place is average about is around 320 km. As a the result, this method reveals approximately the area of $640 km^2$ where the camera is situated. The average error in latitude is 3.5 °and the average longitude error is 2.5 °.

Results are the worst during equinoxes period when it is possible to determine longitude but in latitude the error can be up to 16 °. That is caused by equal lengths of day. We noticed that the location found by our calculations tend to be more to the north from the real location. This is probably caused by error in the astronomical calculations (most probably day saving time) in combination with curvature of the Earth.

Chapter 4

Shadow Analysis

Another direction of approaches uses shadows in the scene to obtain the altitude angle, which leads to determination of the geo-location. In fact, shadows are very important not only as indirect clues for determining the altitude angle, but also for camera calibration.

For human eyes it is relatively easy to find an object and corresponding shadow in the scene. But from the perspective of machine vision the shadow detection and localization is considered as open problem (Section 4.2). Shadows in the scene are used for camera calibration (Section 4.3) and for the Sun's elevation angle measurement by ratio given by shadow length and its corresponding object height (Section 4.4). In our approach, we show possibility of performing the measurements of the shadow length and correspondent object from a single image.

4.1 Method Outline

This method was inspired by traditional navigational tools. The most common tool for determining the position, if we do not include GPS, is a sextant. As we mentioned in Chapter 2 the sextant measures an angle between two celestial bodies. It is used mostly for the measurement of the Sun's altitude angle. With known time and the Sun's altitude angle one can determine the location using the Astronomical Almanac. The first idea was to reconstruct shadow trajectories from the image sequence. We needed to detect only shadows cast by static, non moving objects such as buildings, trees, lamps, monuments etc. This shadow is defined in the time-lapse as an object moving in opposite direction to Sun's apparent movement (the shadow trajectory has direction from the West to the East). This experiment proved to be unsuccessful, due to limitation of methods for shadow detecting and shadow tracking. The second idea relies on the camera calibration.

Shadows in the scene (real world) are parallel in generic situations. This phenomenon is caused because Sun, as the source of light, is considered to be in infinite distance (in fact the mean distance between Earth and Sun is 149597870 km). Shadows in the image are not parallel to each other due to perspective projection. We use this fact for calibration of the camera and recovery of the vanishing lines. Once the camera is calibrated it is possible to reconstruct the real world's coordinate system and consequently measure the angle between the shadow and shadow-casting object. We introduce an approach where it is unnecessary to perform the entire camera calibration. Only two vanishing points are enough for calculation of the altitude angle. Using this approach, under certain circumstances (presence of at least two shadow casting objects), we can measure the altitude angle from a single image only.

4.2 Shadow Detection and Localization

Detection of the shadow in an image is the key step for the method of geo-localization. Most of the approaches for shadow detection rely on the fact, that the shadow in the image can be detected as a pixel with slight difference in the luminescence value. Basic solution is based on the threshold of the luminescence or intensity values. This can be used when we are facing a scene where shadows are sharp and dark, but not in a scene where shadows are barely visible. Obviously, this can cause an error by selecting black coloured object as a shadow.

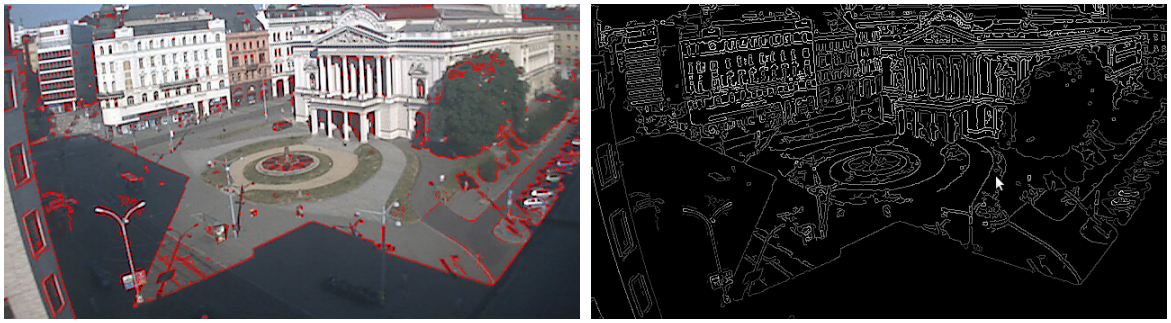
In the beginning of our research our goal was to find a general method for detecting the shadows and shadow trajectories. We developed another approach where only two single shadow lines are required for determining the geo-location. We adapted two methods for shadow detection. In basic principles both work as a motion detector, because in the video shadows are basically moving objects.

In the first method we calculate differences in three consecutive images $(I_1, I_2, I_3)$to create a *Difference shadow image (DSI)* as follows:

$$D = ||I_{1(x,y)} - I_{2(x,y)}| - |I_{2(x,y)} - I_{3(x,y)}|| \tag{4.1}$$

$$DSI_{(x,y)} = \begin{cases} D & \texttt{if} \quad D > T \\ 0 & \texttt{otherwise} \end{cases} \tag{4.2}$$

We can detect borders of moving shadows in the scene in the resulting DSI. Other moving objects e.g. pedestrians, moving vehicles can cause false detection of the shadows. These false shadows are suppressed by using three consecutive images. Two shadows of considerably large objects (Buildings, monuments etc.) will have intersection in two consecutive images. It is highly unlikely for a human to be moving so slow that its silhouette will have intersection in three consecutive images. We are using images captured each 2 minutes from our dataset, which is enough for determining the intersection of moving shadows but omit other objects. The main disadvantage of this method is that for thin shadows from lamps or poles it fails. Unfortunately for estimation of the Sun's elevation angle we need precisely these shadows.



(a) (b)

Figure 4.1: Results of the shadow detector. **(a)** Example of shadow detection. **(b)** The map of shadow edges for the image

In addition of a possibility to detect thin shadows we used the Canny Edge Detector [7]. Whereas the plain Canny Detector finds all edges in the image, with a modification we use, it is possible to find only edges which are located in areas covered by shadows. We achieve

that by masking the edges found by Canny with intensity values of the image followed by thresholding this mask to select only dark edges. Before the masking operation we erode the original image in order to expand the shadow regions. As is shown in Figure 4.1 all the edges in the shadow region are detected. Small regions of dark colours in the images are the most common source of problems for this method. In fact, it is not necessary to find all shadows for the geo-localization but it is important to find thin shadow lines which determine the shadow direction.

The result of this method can be seen in Fig. 4.1a. Red colour highlights edges which belong to the shadowed regions. This result is presented without the elimination of noise shadows. In Fig. 4.1b the map of shadow edges is shown. That is basically classifier for the edges. More darker the edge in the map is, more likely it represents the shadow.

The second method works with a background subtraction. It extracts the moving foreground from the static background. The method works with a Gaussian Mixture-based background segmentation introduced in [21]. It uses a method to model each pixel of the background as a mixture of K Gaussian distributions ($K = 3 \ldots 5$). The mixture represents the time of how long the colours stay in the scene. The colours which stay longer in the scene are selected as the background as a result.

Existing Methods

For the shadow detection and removal, recently Suny and Mihila [35] use a LAB colour space. Using this colour space it is easier to classify the shadow pixels in the image. They remove isolated or false detected pixels by morphological operation. Authors Kumar and Kaur [24] paraphrase various shadow detection methods. Most of them are based on the image segmentation or colour changes of shaded pixels. For the purpose of this work it is necessary to detect shadows in the time-lapse images. The technique for detecting of a shadow trajectory is presented by Wu and Cao [38]. They construct a background image with the brightest pixel. The absolute differencing of a shadow region is specified by thresholding. For determination of the shadow area they select a point in a shadow region and enforce the flood fill algorithm.

There are more and robust techniques for shadow detection. Guo et al. [13] are motivated by regions in the image. They classify regions in the image separately and predict relative illumination for every single region. They perform pairwise classification of regions to determine whether the region is shadow.

Authors Junejo and Foroosh [20] are using relatively simple method based on a background image segmentation. For a set of images $S = \{I_1, I_2, \ldots, I_n\}$they construct a background image with certain properties. This background image $I_{background}$ contains the brightest pixels of images in the image set:

$$I_{background}[x, y] = brightest([x, y](\forall i)) \tag{4.3}$$

where $i \in S$, $[x, y]$ are row and column coordinates of the certain image and $brightest()$ returns the maximum value from intensities. The main disadvantage of this method is that it is semi-automatic, thus the most prominent shadow points must be detected manually and it fails due to certain weather conditions, mostly cloudy weather.

Detection of the Shadow Trajectories

Several experiments were done with shadow tracking methods. After we extract shadow borders by using our shadow detection method we apply a contour detection and Principle

Component Analysis [18] for outlining of the shadows. PCA gives us the direction of the shadow which gives us information for the camera calibration. On the other hand, the determination of the shadow region is not precise enough and the direction from PCA is not always correct.

The crucial task was to find the relationship between two shadows cast by the same object in two images in different time. We chose to use SIFT features to describe shadow regions and Brute-Force matching to find their relationships. SIFT algorithm uses Laplacian of Gaussian to detect blobs that act as features. This method did not work well for finding the relationship between two shadows of the same object in different time. More successful method was to apply another approach using FAST [31] as the feature detector which uses corners and edges in the shadow regions that were extracted from the DIS. Problems within this method occurred when trying to perform matches.

4.3 Camera Calibration

The pinhole camera is a common model describing the relationship between coordinates of real world and its projection onto the image plane. What we need to achieve is a performing measurement in the 3D world (the elevation angle) from a 2D image. To be more precise, we need to find the length of the shadow and its corresponding object in the 3D world coordinates. In order to do that, it is required to understand the perspective projection which describes image formation by a pinhole camera. Parallel lines from the real world do not remain parallel in a perspective image. Usual example is a picture of a railway where two parallel tracks intersects at some point. Figure 4.2 illustrates a basic perspective projection with the display of commonly used terms.

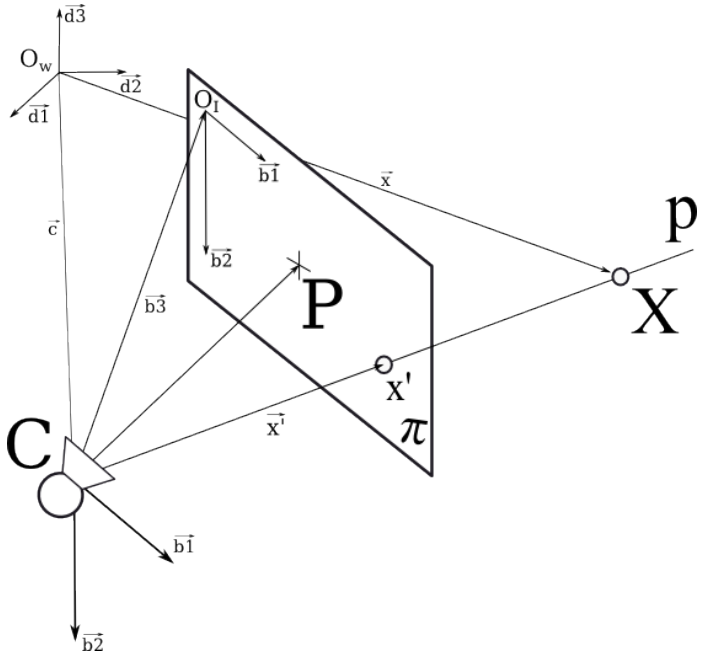


Figure 4.2: Coordinate system of perspective camera model.

Perspective camera model allows us to project 3D world point X into the image plane point x' and find the ray p, projected from 3D space to image plane, along which the point

x' is projected. Ray p is constructed by joining point X with the projection centre C, which is usually the camera. In the model we have three-dimensional space with world coordinate system (O_W, δ), where O_W is the origin and $\delta = [\overrightarrow{d_1}, \overrightarrow{d_2}, \overrightarrow{d_3}]$ is the orthonormal basis. The image coordinate system (O_I, β) looks similar. O_I is the origin and $\beta = [\overrightarrow{b_1}, \overrightarrow{b_2}]$ is the basis of the image coordinate system. In general camera model this basis is not necessary orthogonal. The camera coordinate system $K(O_C, \delta)$ will be derived from the image coordinate system. It has origin in the projection centre C and the basis is $\beta = [\overrightarrow{b_1}, \overrightarrow{b_2}, \overrightarrow{b_3}]$. The third vector $\overrightarrow{b_3}$ which corresponds to vector $\overrightarrow{CO_I}$ is added. Definition of homogenous coordinates and problem of perspective projection is available in [34, 30].

Camera needs to be calibrated when it is used in application which requires interaction with the real word. Calibration itself means to find a set of physical parameters, obtaining a detailed model and description of the world coordinates. The projection of a planar scene by a pinhole camera is related to a homography. The projective transformation of homogeneous coordinates:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4.4}$$

The homography $\mathbf{H_{3\times 4}}$ from the notation above is the projection matrix. It can be decomposed into the product of the camera calibration matrix $\mathbf{K}$ and the transformation matrix from the world to the camera coordinate system $[\mathbf{R}\,\mathbf{t}]$:

$$\mathbf{H} = \mathbf{K}\,[\mathbf{R}\,\mathbf{t}] \tag{4.5}$$

Rotation matrix $\mathbf{R}$ determines the orientation of the camera. The camera calibration matrix does not change when moving its camera in the space. The camera model considers the skew coefficient between two image axes γ, optical centre (p_x, p_y) and the scale factors $\alpha_u f$ $\alpha_v f$. A simplification by taking the skewness to zero and scale factor to equal one is often made. As a result the camera calibration matrix has this form:

$$\mathbf{K} = \begin{bmatrix} f & \gamma & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

where the focal length f is an important parameter. The camera is calibrated when intrinsic and extrinsic parameters are determined. Intrinsic parameters are determined by the calibration matrix and extrinsic parameters are determined by rotation and translation matrices. There are many existing approaches for the camera calibration. For our requirements a method based on calibration from vanishing points [29] is taken into consideration.

4.4 Shadow Measurements

We introduce a solution, where the camera calibration is not necessary. Let us consider a scene containing shadows on a ground plane. In the scene we require a presence of at least two thin objects such as lamps, flagpoles (From now on we will denote them as lamp type objects or simply the *lamp*). Shadows cast by lamps are thin and indicate the direction of

sunbeams falling on the ground. The vanishing point of direction parallel to the shadow directions is considered to be the First vanishing point V_1. As mentioned in the previous section, detection of this type of shadow is a difficult task. Therefore, we label the lamp and its shadow first. In fact, we do not need only regular shaped objects, any shadow can be used. Their shadows to determine the direction and the first vanishing point.

The second Vanishing vector has the direction parallel to the ground plane. We assume that the lamp is pointing up to the sky, therefore its direction is perpendicular to the ground plane. In real life scenarios, it is not common that an object pointing into the sky is perpendicular to the ground plane. Nevertheless, due to the quality of the web camera images, we can assume the direction of the lamp to be perpendicular to the ground plane and determines the second Vanishing point V_2. Moreover for obtaining the second vanishing point we can use other objects. In urban sceneries, where many web cameras are situated, corners of the buildings or similar structures can be used to obtain the perpendicular direction to the ground plane.

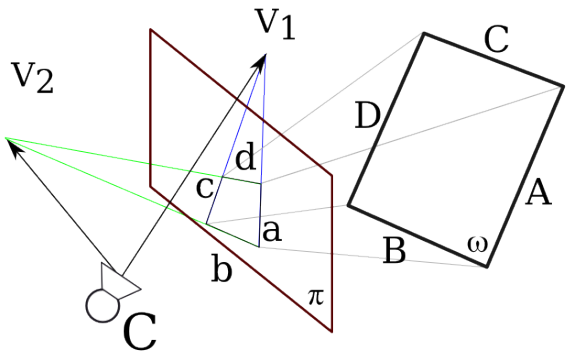


Figure 4.3: Relationship between vanishing points and the plane in the real world coordinates.

The third Vanishing point is not required for our calculations, yet we can calculate its position by using first two VPs. There is a possibility of finding the third Vanishing point directly from the shadows by known length of the day (From the Sunrise and Sunset detection method). In the midday the shadow direction is pointing to the north in the southern hemisphere and to the south in the northern hemisphere. The azimuth angle $a = 0°$ when the shadow is pointing to the north and $a = 180°$ when pointing to the south. We assume knowledge of the day of the year as well. Using astronomical calculations from [11] we can calculate how the azimuth angle changes in each hour and when the angle changes to $a = a \pm 90°$.

Once we have first two vanishing points, the focal legth f can be calculated as:

$$V_1 = (v1_x, v1_y) \;\; V_2 = (v2_x, v2_y) \;\; P = (p_x, p_y) \tag{4.7}$$

$$f = \sqrt{(V_1 - P) \cdot (V_2 - P)} \tag{4.8}$$

We assume the principal point to be in the centre of the image $p_x = \frac{img.width}{2}$ and $p_y = \frac{img.height}{2}$. According to the physical parameters of CCD camera chip the principal point is not necessarily in the centre, although the central principal point is a popular assumption [9].

Without loosing the generality we can assume the camera's position as $C = [p_x, p_y, 0]$ and the image plane π distant f from the camera. The image plane π is described by point $P' = [p_x, p_y, f]$ and vector (the normal vector) $\overrightarrow{CP'}$. Now we have two lamps in the image plane L_1 and L_2. The lamp is described by three points $L = T, H, S$, where T stands for lamp top, H for lamp heel and S for the shadow (Figure 4.4).

The lamp in the image plane is a general triangle T, H, S as a projection from the real world coordinates points T', H', S'. In the 3D coordinate system the $T'H'S'$ it would be a right triangle with a right angle between sides TH and HS. With the information we already have it is possible to transform the 2D triangle THS from the image plane into the real world. We need to recover the the $T'H'S'$ triangle from the THS. The graphical representation of the entire process is shown in the Figure 4.4. First we construct vectors $\overrightarrow{CS}$,$\overrightarrow{CT}$,$\overrightarrow{CH}$ (All points T, H, S lie in the plane π, therefore $S = [S_x, S_y, f] \dots$). The next step is to determine the ground plane in the real world coordinates:

$$\rho = ax + by + cz + d = 0 \tag{4.9}$$

$$\rho = v1_x x + v1_y y + fz + d = 0 \tag{4.10}$$

where the plane (the point-normal general form in the Equation 4.10) is described by its normal vector $\overrightarrow{n} = (n_1, n_2, n_3)$. In our case the normal vector is given by the second vanishing point V_2, hence $\overrightarrow{n} = (v1_x, v1_y, f)$. We can choose the parameter d with only condition which is C does not lie in the ρ. It does not matter where the ground plane is situated. The angle (ratio between two sides of the triangle) must be obtained despite sizes of the triangle. Once we have the ground plane we can extend line segments CH and CS in order to find the intersection with the ground plane.

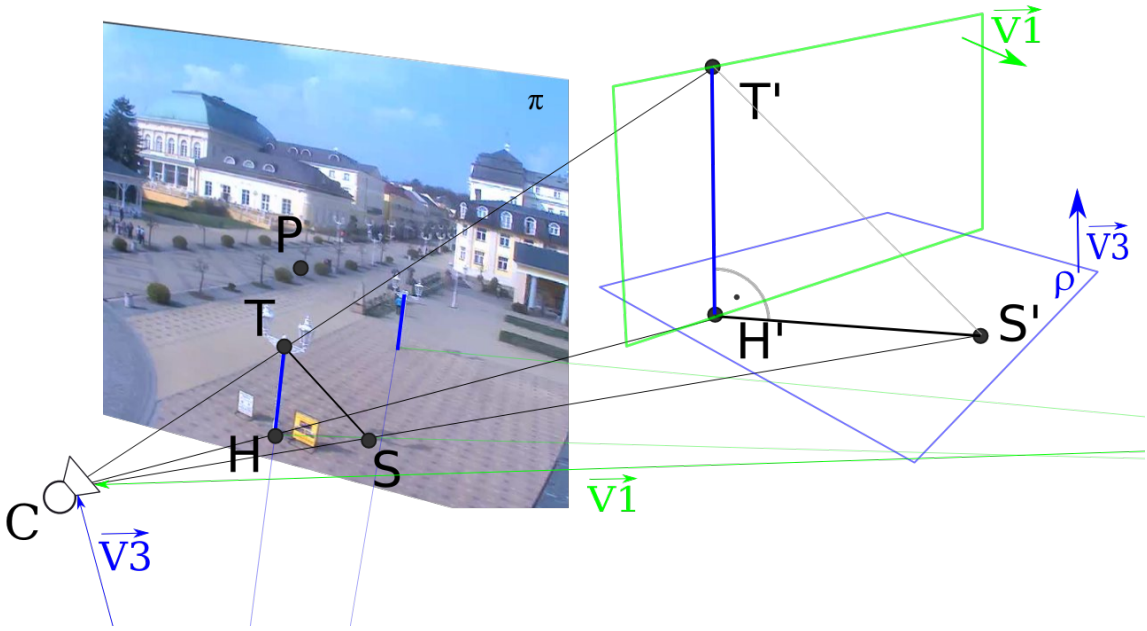


Figure 4.4: The reconstruction of the scene using the camera and vanishing vectors.

$$H' = \rho \cap \overrightarrow{CH} \tag{4.11}$$

$$S' = \rho \cap \overrightarrow{CS} \tag{4.12}$$

The final step is to reconstruct the 3D point T'. The first vanishing point V1 gives the direction of a plane ϱ between two lamps which is perpendicular both to the shadow direction $(S'H')$ and the ground plane ρ. When the position of ϱ plane is known, we can calculate the elevation angle. The plane ϱ has common point with the ground plane ρ and it is described as:

$$\begin{aligned} &\varrho(V_1', H') \\ &V_1' = (v1_x, v1_y, f) \end{aligned} \tag{4.13}$$

Where H' is already known. Finally, we can calculate the elevation angle from the recovered right triangle $T'H'S'$:

$$\alpha = \tan^{-1} \frac{|H'T'|}{|H'S'|} \tag{4.14}$$

We can obtain all angles in the right triangle T', H', S' despite not having a metric to solve the real length of shadow and lamp. Angle between two sides is a ratio regardless we calculate it in meters pixels or any other unit.

4.5 Results and Discussion

We decided to use a semi-automatic method of the shadow detection due to shortcomings in our shadow detection methods. Correct detection of shadows and shadow trajectories seems to be out of the scope of this work. Moreover in requirements for the measuring the altitude angle we need to find correspondence between shadow and the shadow casting object. By Mamassian [26] it is even impossible to find a correspondence between a shadow and corresponding object. Methods we propose are suitable for detecting large shadow areas e.g. shadow of building. Unfortunately shadows that are required for our calculations are typically very thin and very difficult to detect. We experimented with two different approaches, nevertheless neither of them brought accurate results.

The results of the shadow analysis method are very precise. We introduce three cases where we applied proposed algorithm. Each case is represent by table with results Table 4.1 4.2 and 4.3. In the table there is time of the observation, the real Sun's elevation angle and value of the angle measured by our method. Estimated coordinates of observed camera are shown on the bottom of each table.

We can compare results with the previous method. In the first case (in both methods) we are facing the same camera. The sunrise/sunset method estimated the final location 93.11 km from the real place. The shadow measurement method determined the location 61.75 km distant from the physical place of the camera. Moreover, the approximate location of where the camera is situated is more precise using the shadow analysis (Figure 4.6). The Table 4.1 shows how the estimation of the Sun's altitude angle is correct during the course of the day.

In the second case the distance from the camera's location is less precise compared to the first case. It is caused by greater error in the measurement of the Sun's elevation angle.

Figure 4.5: Measured angle $\alpha = 35.16\,°$from the first camera. The orange line is recovered horizon line. The image time is 10:04.

Figure 4.6: The visualization of the results calculated by the shadow analysis method of geo-localization. The red marker is the physical location of the camera, the green marker denotes estimated location. The highlighted are is found by the algorithm.

Location	Date	Time	Real $\alpha_{real}\,[^\circ]$	Detected $\alpha_{detected}\,[^\circ]$	Error $\pm\alpha_e\,[^\circ]$
Františkovy Lázně	26th Mar	9:04	27.22	28.19	0.96
		9:38	31.57	33.37	1.81
		10:04	34.52	35.16	0.64
		10:22	36.34	35.92	0.41
		10:54	39.05	37.85	1.20
		11:16	40.47	38.91	1.56
		13:52	38.1	36.66	1.44
		14:26	34.92	33.22	1.70
		14:52	32.02	30.92	1.10
		15:20	28.53	26.33	2.20
$\varphi = 50.1173\,^\circ$ $\phi = 12.3509\,^\circ$	$\varphi = 50.625 \pm 0.5077\,^\circ$ $\phi_e = 12.0000 \pm 0.3509\,^\circ$ $M_{dist} = 61.75\,\mathrm{km}$				

Table 4.1: Measured angles and detected location for the first case.

The reason why the α is inaccurate is in wrongly determined shadows. In many cases an error of one pixel causes change of an half of arcdegree in the Sun's elevation angle. More distant the vanishing points are, more the error caused by on pixel offset is in effect.

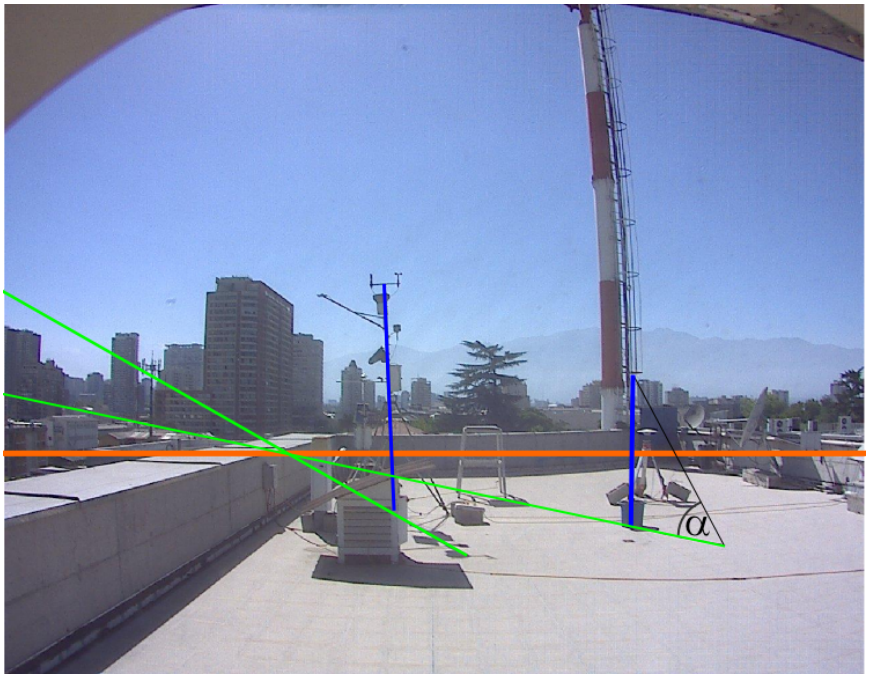


Figure 4.7: The second case scenario. The image time is 20:30.

Location	Date	Time	Real $\alpha_{real}\,[^\circ]$	Detected $\alpha_{detected}\,[^\circ]$	Error $\pm\alpha_e\,[^\circ]$
Santiago de Chile	23th	14:46	28.14	33.9	5.76
	Feb	15:34	37.96	38.12	0.16
	2014	18:36	65.75	68.03	2.28
		19:36	64.5	68.165	3.67
		20:30	57.88	66.21	8.32
$\varphi = -33.45\,^\circ$ $\phi = -70.68\,^\circ$	$\varphi = -30.8 \pm 3.37\,^\circ$ $\phi_e = -71.2 \pm 0.52\,^\circ$ $M_{dist} = 297.92\,\text{km}$				

Table 4.2: Measured angles and detected location for the second case.

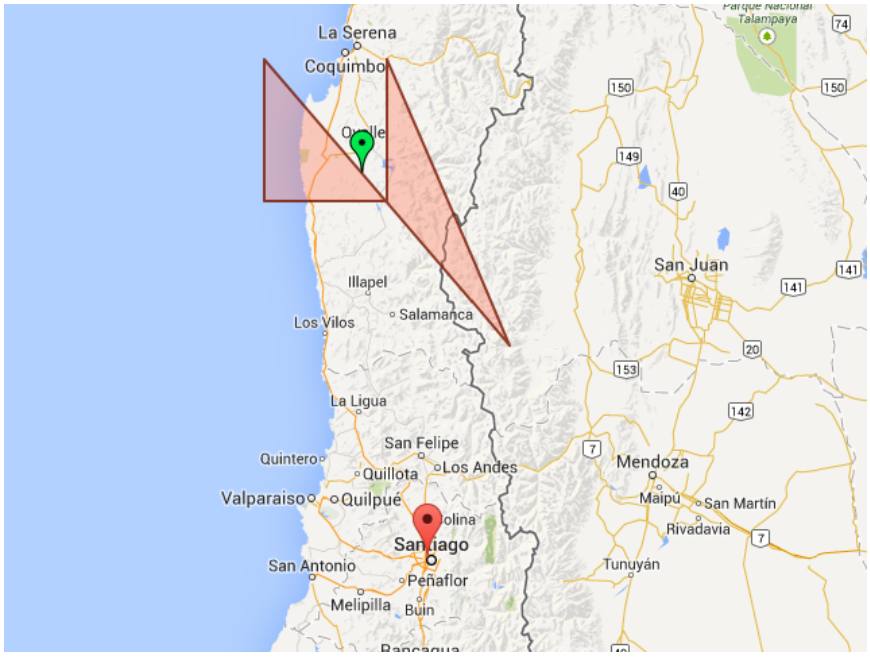


Figure 4.8: The visualization of the secon case - Santiago de Chile.

In the third scenario we calculated the geo-location using only the three measurements of the Sun's elevation angle. The error in the latitude longitude is only 0.01 °and 0.19 °, respectively. The estimated location is only 10.27 km distant from the camera's location.

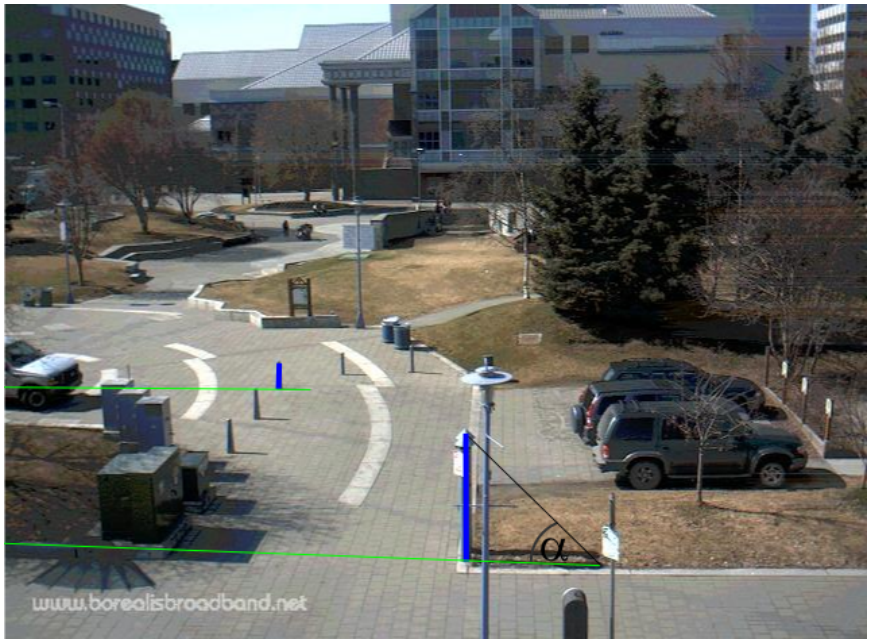


Figure 4.9: The third scenario.

Location	Date	Time	Real α_{real} [°]	Detected $\alpha_{detected}$ [°]	Error $\pm\alpha_e$ [°]
Alaska - Anchorage	1st	0:58	42.51	43.48	0.97
	May	2:12	37.88	40.53	2.65
	2014	4:50	21.21	21.66	0.45
$\varphi = 61.21\,°$ $\phi = -149.89\,°$			$\varphi = 61.2 \pm 0.01\,°$ $\phi_e = -149.7 \pm 0.19\,°$ $M_{dist} = 10.27\,\text{km}$		

Table 4.3: Measured angles and detected location for the third case.

We present a novel method for determination of the Sun's elevation angle using shadows in the scene. This method requires at least two *lamp-type* objects and the flat ground plane to be present in the image. We assume that the lamp-type object is perpendicular to the ground plane. Only one shadow-casting object is required to be fully visible. The other shadow-casting object is necessary for determination of the shadow's direction, but not for the Sun's elevation calculation itself. If the image scene meets this criteria, it is possible to obtain the Sun's altitude angle from a single image. It is not possible to determine the azimuth angle from one image only. Therefore, it is required to have at least two measurements of the elevation angle from two different images. This gives us an indirect information about the azimuth angle between shadows in those two images. On the other

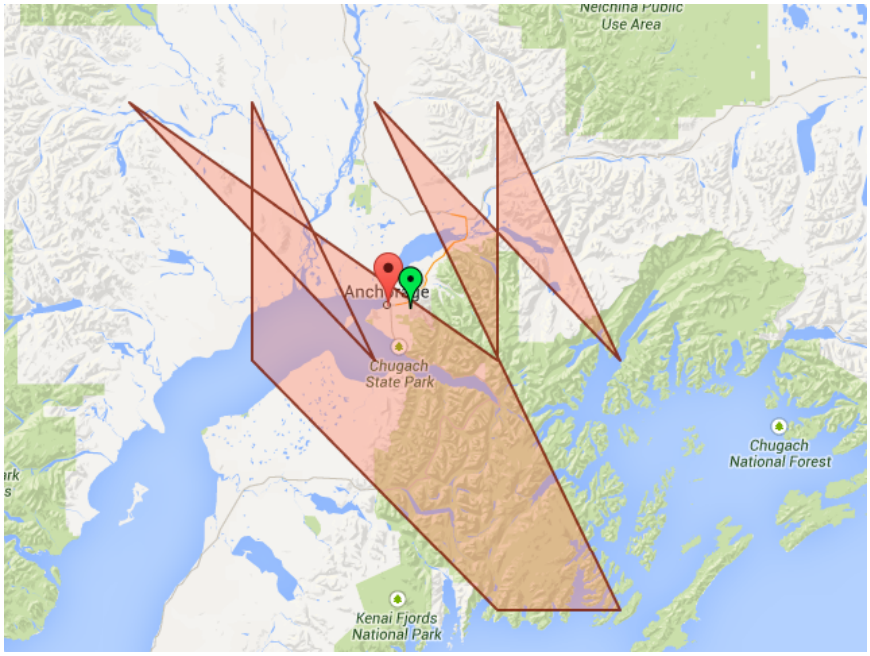


Figure 4.10: The visualization of the third case Alaska, Anchorage - USA.

hand, using only two measurements may lead to ambiguous determination of the location. It is not possible to determine the hemisphere in every situation.

The results of this method are very precise. The estimated location is in the worst case scenario less than 300 km distant from the actual camera's location. Therefore, this method can be used for more precise determination of the geo-location. This method requires presence of shadows in the scene which is not possible all the time. However, using only three measurements of the Sun's elevation angle in the day we can get very accurate precision.

Chapter 5

Astronomy

Astronomical journal and Almanac is the best to consult in situation when we need to know what time will the Sun rises in the certain date. Perhaps some software or a website[1] might help. On the other it is more convenient to develop and adapt calculations of the Sun's position. Calculations are usually simplified, but the results differ from the Astronomical Almanac in less than units of arcminutes.

For understanding the astronomical calculations it is necessary to express time as one unique number related to the apparent Sun's path. Note, that we will use calculations and assumptions where Sun is the reference point, but in general we can use any other star. A calendar helps people to keep track of time. One day is approximately one full rotation of the Earth around its axis. One month is is the time taken by Moon to complete one full circle around the Earth. A year is the time taken for the Earth to complete one circuit around Sun. The problem with this method lies in the fact that there is always a whole number for days in the civil year, Earth takes about 365.2422 days to complete one full circle around the Sun. Consequently, after 100 years we can get error about 24 days. Clearly this system have many disadvantages. During the centuries mankind changed calendar types several times. Julius Caesar introduced the Julian calendar it worked for centuries until 1582, when there was and appreciable discrepancy between seasons and the date cased by missing out three days every four centuries. Therefore Pope Gregory XIII abolished 10 days and introduced some reforms of the calendar. A new system is called the Gregorian calender and is the one in general use today.

For astronomers it is necessary to express a date as a fraction of a day after a given fundamental epoch. Astronomers have chosen Greenwich mean noon (midday on the Greenwich meridian) of 1 January 4713 BC. The number of days elapsed since that is known as the Julian day number, or Julian date. Similarly to Unix time which is a system describing time as the number of seconds since *the beginning of the epoch* 1 January 1970 (00:00:00 UTC). Unix epoch time in JD is equal to 2440587.5. There are some specific condition in the Julian date system that can be found in [11].

The geographical location is typically given by two numbers: longitude and latitude. There are several coordinate systems. The conventional one, we use the most, is called the Equatorial system. In astronomical calculation another system called the Ecliptic coordinate system is used. Each system includes in the name the fundamental plane which it uses as a reference. The ecliptic system makes all measurements with respect to the plane of the ecliptic, the plane of the Earth's orbit around the Sun. This is convenient for representing

[1] http://suncalc.net

the Sun's apparent position as seen from the Earth. We will not go deeper into explanation of the conversions, aberration, atmospheric refraction etc. All mentioned problems are described in [3]. The algorithm has precision of $0°0'36''$ for the period dated 1950 - 2050.

5.1 Sunrise Sunset

Sunrise occurs when the Sun's upper rim appears above the horizon. Finding the time of sunrise is basically finding the time when the altitude angle is zero. The Sun's apparent radius at the horizon is 16 arcminutes and the refraction caused by atmosphere is 34 arcminutes. Combining those two angles the sunrise occurs when the centre of the Sun is 50 arcminutes below the horizon (zenith anlge is $90°5' = 90.8333°$). During the course of a year Earth moves in elliptical orbit around the Sun. Viewed from Earth, it seems that the Sun is moving around Earth, hence we assume that Earth is static and the Sun describes an ellipse around Earth which is convenient assumption in the calculation of the Sun's apparent position.

Based on the given location we calculate when the sunrise occurs (More precise explanation of each step is described in [11]). Inputs for the algorithm are: date of sunrise/sunset, latitude φ and longitude ϕ, Sun's zenith angle ($z = 90.8333°$ for the official sunrise). In the first steps we calculate the day of the year as DoY, convert the longitude to the hour angle HA and to find an approximate time of the sunrise/sunset t:

$$HA = \frac{\phi}{15} \tag{5.1}$$

$$t = DoY + \frac{s - HA}{24} \tag{5.2}$$

Where $s = 6$ or $s = 18$ depending whether we want to calculate the sunrise time or the sunset time. Next we calculate the Sun's mean anomaly M as:

$$M = (0.9856t) - 3.289 \tag{5.3}$$

We can proceed to determining the Sun's true longitude ϕ_{sun} (Note, that ecliptical coordinates are in use).

$$\phi_{sun} = M + (1.916\ \sin M) + (0.020\ \sin(2M) + 282.634 \tag{5.4}$$

Once we know the ϕ_{sun} it is important to find the Right ascension of Sun RA which needs to be in the same quadrant as ϕ_{sun}:

$$RA = \tan^{-1}(0.91764\ \tan\phi_{sun})$$
$$RA = \frac{RA}{15} \tag{5.5}$$

The final step is to find the Sun's local hour angle HA_{sun} and declination angle δ. In this step the zenith angle takes an effect:

$$\delta = \sin^{-1}(0.39782\ \sin\phi_{sun}) \tag{5.6}$$

$$HA_{sun} = \cos z - \frac{\sin\delta \sin\varphi}{\cos\delta \cos\varphi} \tag{5.7}$$

From the HA_{sun} the mean time of rising/setting is calculated:

$$\begin{aligned} \texttt{if sunrise} : H &= 360 - \cos^{-1} HA_{sun} \\ \texttt{if sunset} : H &= \cos^{-1} HA_{sun} \end{aligned} \tag{5.8}$$

$$Time = H + RA - (0.06571 * t) - 6.622 - HA \tag{5.9}$$

The representation of the resulting $Time$ is in the Universal Time format, therefore it needs to be converted to the local time zone by local time offset. This algorithm is general algorithm to determine the time of sunrise/sunset given by the geographic position and date. For determining the geo-location we need to adapt this algorithm in the form where input is the T_{set}/T_{rise} and date. Solving the Equation 5.7 for φ, to determine the latitude results in:

$$\varphi = \tan^{-1} \frac{\cos\delta(\cos z - HA_{sun})}{\sin\delta} + \pi n,\ n \in \mathbf{Z} \tag{5.10}$$

Unfortunately this equation must meet the conditions $(\cos\delta)^2(HA_{sun} - \cos z)^2 + s^2 \neq 0$ which is not possible for many cases, due to the fact how is the δ declination calculated ,also the calculation does not work for cases where $\delta = 0$. Therefore we adapter numerical approach. Our method calculates appropriate T_{rise} and T_{set} for every longitude and latitude in given date. From all results we find the one with the best approximation to the values we obtained from the visual analysis of the scene. The similar principle is applied in calculation of the position based on the elevation angle.

5.2 Elevation Angle

The position of the Sun as seen by an observer from the Earth is most commonly described in equatorial coordinates. By the azimuth angle and the altitude angle. We use the same assumption as in the algorithm for calculating the T_{set}, T_{rise}. Moreover we define a plane which contains the Sun-Earth orbit as the plane of the ecliptic. There is no need to worry about deviations from the ecliptic, because the Sun's position as the ecliptic latitude is zero all the time.

First we define the epoch on which our algorithm is based (1st January 2010 is usually chosen). The Sun's mean ecliptic longitude at the epoch is $\epsilon_{lng} = 279.6966778°$. This number is approximation of a circular orbit. In reality the orbit is an ellipse , therefore we use ϵ_{lng} as the starting point. To make the orbit elliptical we need two other constants $\omega_{lng} = 281.2208444°$, the longitude of the Sun at perigee (Time when Sun is the closest to Earth. The opposite is apogee when Sun is the farthest), and $e = 0.01675104$, the eccentricity of the Sun-Earth orbit. From now on we assume that Sun moves around Earth at a constant speed, rather than along the ellipse. We can calculate the Sun's mean anomaly M which is correction of assumed circular trajectory by including perigee:

$$M = \frac{360}{365.242191} D + \epsilon_{lon} - \omega_{lon} \tag{5.11}$$

Where ϵ_{lng} and ω_{lng} is the mean longitude of Sun at the epoch and at perigee, respectively. The constant 365.242191 tells in how many days the Sun completes the circle of

360°. This leads to better understanding of M from previous algorithm. The key variable in this formula is D the number of days since the epoch. For the true trajectory of Sun as an ellipse we need to obtain the true mean anomaly M_{true}:

$$M_{true} = M + \frac{360}{\pi} \sin M \tag{5.12}$$

Having the M_{true}, we only add ω_{lng} to get the longitude of Sun:

$$\phi_{sun} = M_{true} + \omega_{lng} \tag{5.13}$$

When the ϕ_{sun} is known the right ascension and declination can be found. Consequently from RA and δ we can obtain the altitude and azimuth angle which is the position of Sun for observer on Earth given by D. The ecliptic latitude of the Sun $\varphi_{sun} = 0$ because the Sun is in the ecliptic plane conversion into RA and δ is done using the formulas:

$$RA = \tan^{-1} \frac{\sin \phi_{sun} \cos O - \tan \varphi_{sun} \sin O}{\cos \phi_{sun}} \tag{5.14}$$

$$\delta = \sin^{-1} \sin \varphi_{sun} \cos O + \cos \varphi_{sun} \sin O \sin \phi_{sun} \tag{5.15}$$

Where the O is the obliquity of the ecliptic, the angle between the planes of the equator and the ecliptic, in the other words it is a conversion factor between conversion from elliptical coordinates into equatorial coordinates. The O is represented in the Greenwich calendar date, but it is changing so slowly that a value from the middle of given years can be used as a constant. The last step is to convert equatorial coordinates given by HA, which is calculated from RA, and δ into horizon coordinate system using the azimuth A and the altitude angle α:

$$\alpha = \sin^{-1}(\sin \delta \sin \phi + \cos \delta \cos \phi \cos HA) \tag{5.16}$$

$$A = \cos^{-1} \frac{\sin \delta - \sin \phi \sin \alpha}{\cos \phi \cos \alpha} \tag{5.17}$$

For determining the geographical coordinates of a place based on α we use a similar approach as in the previous method. In fact it is possible to find approximate location from a single image. To measure the altitude angle we use method described in 4.4. The information about the azimuth A is not always possible to obtain from the shadow analysis. The azimuth angle $A = 180°$ at the midday. The approximate location can be determined with error lesser than 1°of the latitude and longitude assuming the azimuth angle greater or lesser than 180 depending on the time when was the image taken. The midday can be determined by dividing the length of the day or from the local time of the image. The other possibility to obtain the coordinates is using two images with the considerable difference in the time-stamp (usually more than one hour is enough). The difference in altitude angles and the offset between them are enough to avoid ambiguity of the same altitude angle in the two different locations. Obviously more altitude angles we measure and use for the calculation, more precision the estimation of the location is.

5.3 Twilight

Twilight is the period of transition between day and night. From physical point of view it is typical illumination of the horizon caused by the Sun's position below the horizon and

sunlight scattering in the atmosphere. The length of the twilight differs during the course of the year depending the latitude of place on Earth. In the equatorial areas the twilight lasts around 70 minutes and during the year is is about the same length. The twilight lasts more, more furtherer from the equator is a observer.

There are three different types of the twilight. Each one of them is important either for navigation purposes or our methods of geo-localization. We will discus every type of twilight and explain why it is important for our method. The twilight types are order chronologically how it goes during the daybreak.

- Astronomical twilight: When $\alpha = -18°$ the astronomical twilight is defined. During the Astronomical twilight is possible to observe stars with naked eye, but only in some places. Due to high level of light pollution in urban areas stars might not be visible to naked eye. Astronomical twilight is nor important for our method the sky is very dark during this period and it is hardly recognisable from the night. Moreover observation of stars is not possible with a simple web camera.

- Nautical twilight: Begins when $\alpha = -12°$. Nautical twilight is very important for seafarers to determine location. Sky is dark enough to observe stars, but on the other hand bright enough for horizon to be apparently visible. Nautical twilight is important for detection of time when night changes into day and other way around. For observer on the Earth's surface during the Nautical twilight colour of the sky goes from dark blue to bright blue, therefore horizon is starting to be clearly visible. The end of the nautical twilight is often detected by our method as

- Civil twilight: Begins when $\alpha = -6°$. During the civil twilight artificial lights are no longer required in order to properly see for human eye. The brightest star in the sky, Venus is visible. This period is proceed by the sunrise and consequently by day. The end of the civil twilight is usually detected by our method as the beginning of the day in the most cases.

The sunset happens when α, solar elevation angle equals zero. That would mean that centre of the Sun is aligned with the horizon. As mentioned before the α of sunrise is -0.8333°, due to refraction in the atmosphere.

5.4 Precision of Calculations

One arcdegree of the latitude is approximately 111 km distance on the Earth's surface. The distance between two places with the difference of 1 arcdegree of latitude varies due to Earh's ellipsoid shape from 110.6km at the equator to 111.7 at the poles. A degree of the longitude gives distance of 111.3km at the equator and it proportionally decreases to poles where this distance is 0km. For example the distance between places $P_1(\varphi = 50, \phi = 0)$ and $P_2(\varphi = 50, \phi = 1)$ is 71.7km. The error of one arcdegree in the calculation may lead to the 111km deviation from the position of the actual place. Date we use for the detection of the sunrise and sunset are sampled each 2 minutes. Deviation of 2 minutes in the T_{rise} ($T_{error} = 2\,\text{min}$) from the actual sunrise time causes an error of $\varphi_{error} = 0.5\,°$and $\phi_{error} = 1°$. This error rate does not grow linearly for example if the $T_{error} = 20\,\text{min}$, the $\varphi_{error} = 3.6\,°$ and $\phi_{error} = 2.6\,°$. The error in the declination angle $\delta_{error} = 1\,°$ causes $1\,°$error in the resulting latitude because the declination angle and latitude are directly dependent.

Chapter 6

Datasets

This chapter gives more detailed description of the images and image sequences which were used for our experiments. There are two different methods of estimating the position discussed in this document. The first method is based on classifying the image sequences into day or night time image was described in Chapter 3. The second approach is based on the shadow analysis of the scene (Chapter 4).

6.1 Images of Day and Night Time

As was mentioned before, for the detection algorithm we need a training set of images labelled into two categories. For purposes of this work the dataset of 350 images of day and 350 images of the night were collected. The main sources are free images found on the internet, from public accessible web cameras or personal photos of the author. Some images were taken from free image databases[1]. For experimenting of twilight detection 100 photos of dusk and dawn are present in the dataset as well. It is required that all phenomenons described in Section 3.4 are present in the dataset. Although the dataset should be robust enough, mostly images of urban sceneries and architecture are present in the dataset. Nature or landscape sceneries are not appropriate, especially for night images where only black colour is present.

6.2 Images from Web Cameras

For the testing purposes methods of the geo-location estimation another dataset was created. We present a collection of the data (image sequences) collected from web cameras. We use image sequences because we do not save the entire video record from each camera but only certain snapshots. Storing videos from a camera uses vast of hard drive capacity, after the set of experiments we decided to save snapshots from the webcamera in the frequency of every two minutes. This gives a balance between disk usage and satisfactory precision of the methods. However from 10 cameras we have samples in the frequency of every minute. This gives us relevant information how the precision of methods change if we can have more frequent image data.

Each image is timestamped with the time when the image was saved. Time format of the time-stamp is `GMT+1`, therefore we don't know the local time of the camera. In those cases where the knowledge of local time is necessary, we can convert to local time. Either based

[1] `imageafter.com`, `freeimages.com`, `openphoto.net`

on the detected times of sunrise or sunset or with known location. But in fact it is suitable that we are using one global time, because all calculations work with the Universal Time (UT). The mentioned cameras are also annotated with the location, or at least estimation of the location in cases where camera's position is unknown.

80 cameras from all over the world were selected to save image information and experiment with them. Unfortunately the impact of technology development makes more difficult to find web cameras in certain regions. The most of web cameras available online are situated in North America, Europe or Japan. For the best observation the equally distributed cameras all over the world would be ideal. It is very hard to find a camera giving data in suitable quality from regions in equatorial and Tropics of Cancer and Capricorn.

Chapter 7

Implementation

The implementation of the suggested methods: Day and night detection and Shadow angle calculations described in Chapter 3 and 4 respectively, were mainly written in C++ programming language.

- **OpenCV** [6] - is a commonly used library for computer vision and image processing. Its constantly updating development make this library very useful tool with many implementations of computer vision algorithms available and ready for use. We found usage of this library for working with ANN, SVM, SIFT, FAST etc.
- **Boost** - a set of C++ libraries working on almost any modern operating system. It contains many single libraries and implementation of lots of useful generic algorithms which are not included in standard libraries. Helpful for working with command-line arguments and file system structure for opening images.
- **Python** - is chosen as a general-purpose powerful and very fast programming language for performing the astronomical calculations. Those calculation are important for determining the location from time of sunrise and sunset or from the altitude angle. Its benefit is a simple creation of prototype program within a few lines of code. The main benefit was usage of `Daytime` module for working with dates.
- **GNUPlot** - is a useful visualisation and plotting tool used in many scientific fields. We found usage of gnuplot mostly to visualize our results into graph, during experimenting with classification of features for machine learning. It was very beneficial especially for visualization of camera calibration and mapping the image perspective into real worlds coordinate system with calculation of shadow angle.[1]
- **QT** - is mostly known as a UI framework and cross-platform development tool. However QT contains the `QTCreator` which could be use as a powerful standalone IDE. Source codes are easy to compile and portable[2] with usage of the `qmake` routine. QT was also used for the visualisation and experimenting with the classification algorithm.

[1] Gnuplot script is one of output from the camera calibration and calculation of altitude angle. Script gives easier view on the projection of entire scene.

[2] Source codes of the implementation and experiments are available on the supplemental CD.

Chapter 8

Conclusion

The goal of the Master's thesis was to study the existing methods of camera geo-localization, to suggest a new strategy and to explore visual clues in the scene leading to determining the location of the camera. The geo-localization methods based on shadow detection were in the aim of the study. During the experiments we found another method, based on sunrise and sunset times, which had considerable success and less problematic development compared to shadow-based method. The primary objective of the work was to reach the accuracy of 200 km. The difference between the actual position and our first method output is on average 320 km. Although the goal was not reached, without prior knowledge of the problem it was difficult to make a guess to how precise the method could be. The second method shows better results. It is possible to estimate the physical location of given camera with the deviation of 175 km on average.

Two different approaches were proposed. One is fully automatic and the other one is semi-automatic and needs manual input of certain data.

A novel method for geo-locating a web camera based on analysing sunset and sunrise time within the image lapse section is suggested. Its main benefit over the other casual methods is that it doesn't required perfect weather conditions in the scene. It is suitable basically for any available outdoor web camera. The method determines the location of camera with precision in range: 100 - 1000 km. Part of this method, namely the day/night classification can be used in automatic management and organisation of photos and multi-media collections.

The second method is derived from the position of the Sun. The Sun is usually not visible in the scene but our semi-automatic method uses shadows as an indicator of the solar light source. Moreover shadows in the scene can be used for determining vanishing points, horizon line and calibrating the camera. Our contribution is in the possibility of estimating the Sun's elevation, given by the altitude angle, form a single image. This method only requires presence of two shadow casting object in the image scene. With this method an altitude angle accuracy of about 4 °was achieved. That is enough for the correct estimation of the location with a deviance of less than 300 km.

Problems are discussed within both methods. In the first method a correct estimation of precise time of sunrise or sunset is problem. The latter method is more problematic. We experimented with two different approaches of shadow detection and shadow tracking. Neither of them was suitable for our implementation. When the problem with shadow detection and shadow tracking is solved, a fully automatic geo-location can be achieved. The method fails due to limits brought by shadow detection algorithm. In some cases the quality of the images from camera is the source of a problem.

Furthermore, for the propos of experiment and evaluation of the method a large amount of data was acquired. Data from public accessible web cameras are a good resource for evaluating or testing computer vision algorithms.

8.1 Future Work

The main limitation for the shadow-based geo-localization techniques is in insufficiency of methods for shadow detection and localization. We used shadow detection method based on motion detector and background extraction which usually failed when lighting conditions in the scene changed. It is important to determine the direction of the shadow for the correct geo-localization. This shadow direction is represented in the scene as a relatively thin shadow line. Therefore an algorithm for detecting shadow lines is a key element for shadow based geo-localization. In addition an automatic shadow tracking algorithm needs to be created for the fully automatic solution of geo-localization. To develop tracking algorithm, the problem of descriptors and features should be studied in more details. Many open challenges can be found in the field of shadow analysis, as we already mentioned.

Besides that the different visual information can be used for estimating the geo-location including type of climate, vegetation or detection and classification of several objects. Combination of both proposed methods with any other geo-localization algorithm can lead to better results.

Bibliography

[1] Global positioning system [online]. [cit. 2014-01-09].
Available: http://en.wikipedia.org/wiki/Global_Positioning_System.

[2] Noaa geostationary satellite server [online]. [cit. 2014-01-09].
Available: http://www.goes.noaa.gov/.

[3] *Almanac for Computers, 1990.* U.S. Government Printing Office, 1990.

[4] Exchangeable image file format for digital still cameras: Exif version 2.2, 2002. Standard of Japan Electronics and Information Technology Industries Association.

[5] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2009.

[6] Dr. Gary Rost Bradski and Adrian Kaehler. *Learning Opencv, 1st Edition.* O'Reilly Media, Inc., first edition, 2008.

[7] J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8:679–698, November 1986.

[8] Eric Cheng, Eric Shrader, and Egbert Tse. Automatic camera geo-location from solar shadows. Available: www.ipam.ucla.edu/programs/rips2011/arete_project.pdf, 2011. Research in Industrial Projects for Students 2011.

[9] T. A. Clarke, X. Wang, and J. G. Fryer. The principal point and ccd cameras, 1998.

[10] Fabio Cozman and Eric Krotkov. Robot localization using a computer vision sextant. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 106–111, May 1995.

[11] P. Duffett-Smith and J. Zwart. *Practical Astronomy with your Calculator or Spreadsheet.* Cambridge University Press, 2011.

[12] William C. Forsythe, Edward J. Rykiel Jr., Randal S. Stahl, Hsin i Wu, and Robert M. Schoolfield. A model comparison for daylength as a function of latitude and day of year. In *Ecological Modelling*, volume 80, pages 87 – 95, 1995.

[13] Ruiqi Guo, Qieyun Dai, and Derek Hoiem. Single-image shadow detection and removal using paired regions. In *CVPR*, pages 2033–2040, 2011.

[14] H. Y. Hseih and N. Chen. Recognising daytime and nighttime driving images using bayes classifier. In *IET Intelligent Transport Systems*, 2010.

[15] N. Jacobs, S. Satkin, N. Roman, R. Speyer, and R. Pless. Geolocating static cameras. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–6, 2007.

[16] Nathan Jacobs, Mohammad Islam, and Scott Workman. Cloud motion as a calibration cue. In *Conference on Computer Vision and Pattern Recognition(CVPR)*, 2013.

[17] Nathan Jacobs, Kylia Miskell, and Robert Pless. Webcam geo-localization using aggregate light levels. In *WACV*, pages 132–138. IEEE Computer Society, 2011.

[18] Dong Hyun Jeong, Caroline Ziemkiewicz, William Ribarsky, and Remco Chang. Understanding principal component analysis using a visual analytics tool.

[19] Imran N. Junejo and Hassan Foroosh. Using solar shadow trajectories for camera calibration. In *ICIP*, pages 189–192. IEEE, 2008.

[20] Imran N. Junejo and Hassan Foroosh. Gps coordinates estimation and camera calibration from solar shadows. *Computer Vision and Image Understanding*, 2010.

[21] P. Kaewtrakulpong and R. Bowden. An improved adaptive background mixture model for realtime tracking with shadow detection, 2001.

[22] Ethan Katz-Bassett, John P. John, Arvind Krishnamurthy, David Wetherall, Thomas Anderson, and Yatin Chawathe. Towards ip geolocation using delay and topology measurements. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, IMC '06, pages 71–84, New York, NY, USA, 2006. ACM.

[23] Byung Soo Kim, Jae Young Park, Anush Mohan, Anna Gilbert, and Silvio Savarese. Hierarchical classification of images by sparse approximation. In *Proceedings of the British Machine Vision Conference*, pages 106.1–106.11, 2011.

[24] Sanjeev Kumar and Anupreet Kaur. Algorithm for shadow detection in real colour images. In *International Journal on Computer Science and Engineering*, volume 2, 2010.

[25] Gibbs S. L. Sundials: History, theory, and practice. by rene j. rohr. *American Journal of Physics*, 43:116–116, jan.

[26] P. Mamassian. Impossible shadows and the shadow correspondence problem. *Perception*, 33:1279–1290, 2004.

[27] W.E. May. *A History of Marine Navigation*. Norton, 1973.

[28] U.s. Nautical Almanac Office. *Astronomical Almanac for the Year 2013 and Its Companion, the Astronomical Almanac Online*. Astronomical Almanac For the Year. U.S. Government Printing Office, 2012.

[29] R. Orghidan, J. Salvi, M. Gordan, and B. Orza. Camera calibration using two or three vanishing points. In *Computer Science and Information Systems (FedCSIS), 2012 Federated Conference on*, pages 123–130, Sept 2012.

[30] Tomas Pajdla. *Geometry of Computer Vision*. FELK, CVUT, 2012.

[31] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06, pages 430–443, Berlin, Heidelberg, 2006. Springer-Verlag.

[32] Steven W. Running and Joseph C. Coughlan. A general model of forest ecosystem processes for regional applications i. hydrologic balance, canopy gas exchange and primary production processes. In *Ecological Modelling*, volume 42, pages 125–154, 1988.

[33] Nawar S., A. B. Morcos, and J. S. Mikhail. Photoelectric study of the sky brightness along sun's meridian during the march 29, 2006 solar eclipse. *New Astronomy*, 12:562–568, oct 2007.

[34] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, 2007.

[35] Ashraful Huq Suny and Nasrin Hakim Mithila. A shadow detection and removal from a single image using lab color space. In *International Journal of Computer Science Issues (IJCSI)*, volume 10, 2013.

[36] Aditya Vailaya, Mario A. T. Figueiredo, Anil K. Jain, and HongJiang Zhang. Bayesian framework for semantic classification of outdoor vacation images. In *Proc. SPIE*, volume 3656, pages 415–426, 1998.

[37] Yanni Wang and Bao-Gang Hu. Hierarchical image classification using support vector machines. In *The 5th Asian Conference on Computer Vision*, 2002.

[38] Lin Wu and Xiaochun Cao. Geo-location estimation from two shadow trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 585–590, 2010.

Appendix A

Content of Supplemental CD/DVDs

- `src/`: Source codes of our implementations, astronomical calculation scripts, experimental programs.
- `doc/`: The technical report including LaTeXsources.
- `README`: A text file with instructions about the dataset.
- `poster, video`: A poster and a video presenting the work. item `dataset/`: The dataset which was acquired by the purpose of this work.