

Are You Lying: Validating the Time-Location of Outdoor Images

Xiaopeng Li, Wenyuan Xu^(✉), Song Wang, and Xianshan Qu

Department of CSE, University of South Carolina, Columbia, SC, USA
{x14, wyxu, songwang, xqu}@cec.sc.edu

Abstract. Photos have been commonly used in our society to convey information, and the associated contextual information (i.e., the capture time and location) is a key part of what a photo conveys. However, the contextual information can be easily tampered or falsely claimed by forgers to achieve malicious goals, e.g., creating fear among the general public or distorting public opinions. Thus, this paper aims at verifying the capture time and location using the content of the photos only. Motivated by how the ancients estimate the time of the day by shadows, we designed algorithms based on projective geometry to estimate the sun position by leveraging shadows in the image. Meanwhile, we compute the sun position by applying astronomical algorithms according to the claimed capture time and location. By comparing the two estimations of the sun position, we are able to validate the consistency of the capture time and location, and hence the time-location of the photos. Experimental results show that our algorithms can estimate sun position and detect the inconsistency caused by falsified time, date, and latitude of location. By choosing the thresholds to be 9.2° and 4.8° for the sun position distance and altitude angle distance respectively, our framework can correctly identify 91.1% of the positive samples, with 7.7% error in identifying the negative samples. Note that we assume that the photos contain at least one vertical object and its shadow. Nevertheless, we believe this work serves as the first and important attempt in verifying the consistency of the contextual information only using the content of the photos.

Keywords: Capture time and location · Sun position · Shadows · Consistency · Projective geometry

1 Introduction

Benefiting from the development of digital technologies and internet, photos become increasingly common in our society. A huge number of photos are shared through social media platforms. People use photos to convey information and express emotions, and even employ them to illustrate news stories [14]. Meanwhile, people are exposed to fake photos that had been used for malicious purposes: fooling the world and creating chaos as well as panic [3, 7]. For example,



Fig. 1. A photo that was taken in Sept. 2013 was used for a news event happened in Jan. 2017.

the Hurricane Sandy hit the northeastern U. S. in 2012: numerous fake disaster photos and rumors were spread through social networks and caused panic and fear among the general public [10]. Therefore, the U.S. Federal Emergency Management Agency had to set up a “rumor control” section to defend against misinformation including fake photos on social networks [1]. In addition, fake photos have been used to distort public opinions. For instance, the fake refugee photos were shared online in the Europe’s refugee crisis in 2015 and used to twist public opinion on asylum seekers [5].

Previous studies mainly focused on devising forensic techniques to detect photo tampering and manipulation. For example, researchers have proposed approaches to demonstrate copy-move manipulation [2, 8, 23] and leveraged shadows and lighting to determine photo tampering [4, 20]. However, in addition to manipulating the content of a photo itself, the contextual information (i.e., the capture time and location) can also be falsified. For instance, the photo in Fig. 1 was claimed to be taken in January 2017, and was used on social medias to illustrate the news that a fleet of bikers were on the way to Washington D.C for President Trump’s inauguration. However, the photo was actually published in 2013 for the anniversary of 9/11¹. Thus, it is promising if we can validate the capture time and locations immediately purely using the photos themselves.

Determining whether the capture time or the location of an image is real is promising yet challenging. Although most images have timestamps and GPS information enclosed, these can be altered without traces once the format is known. Deciding whether a picture is taken at a place simply by experiences is infeasible since the image scenes may appear to be similar in various places, such as public lawns, parking lots, beaches and roadsides. Even if the capture location is true, the capture time can be falsified without any traces. Finding evidences from the content of an image to verify the time is difficult. Objects that reveal time directly (e.g., clocks and watches) are rarely seen in images.

¹ <https://www.buzzfeed.com/tasneemnashrulla/bikers-for-trump-inauguration-fake-pictures>.

Objects such as clothing, or colors of trees may indicate the capture time, but these indicators can only reveal a relatively long time span (e.g., a T-shirt is suitable from April through October in many places). So far, limited research has addressed this problem. Garg *et al.* [9] demonstrated the feasibility of using the Electric Network Frequency signal as a natural timestamp for video data in an indoor environment. Junejo and Foroosh [17] and Wu and Cao [27] used shadow trajectories to estimate the geo-location of stationary cameras from multiple outdoor images. Tsai *et al.* [26] and Kakar and Sudha [18] developed approaches that leverage the geolocation of images and the sun information to estimate the capture time for outdoor images. However, to the best of our knowledge, none has been done to validate both the capture time and location. In this paper, we study how to validate whether the image’s capture time and location are true from a single outdoor image that has at least one shadow. Although we require a shadow in an image, we believe our work serves as the first attempt towards a full-fledged solution.

The basic idea is that the position of the sun is determined by time and location and can be utilized to check time-location consistency of outdoor images. Specifically, we estimate the sun position from two sets of information: (1) utilize vertical objects and their shadows in images to estimate sun position, and (2) use the claimed capture time and location in the metadata of images for estimation. Finally, we compare these two values and decide whether the claimed capture time and location are true.

In summary, we outline our three main contributions as below:

- We propose a framework that is called AYL for validating time-location consistency of outdoor images. We show that the variances of sun position correlate with the time and location, and the correlation can be used to determine whether the capture time and location of images are consistent.
- We demonstrate that the sun position can be acquired from shadows and design algorithms to estimate the sun position from one vertical object and its shadow in the image. The results show that the algorithms are effective.
- We implement the proposed framework and evaluate it using photos collected in 15 cities across the U.S. and China, which proves AYL to be effective.

2 Overview

We specify the threat model, overview the framework of AYL, and summarize the research challenges in this section.

2.1 Threat Model

We assume that an attacker modifies the capture time and location of an image for malicious purposes, but doesn’t tamper or manipulate the image itself. Note that even if she modifies the image, we can detect it utilizing the prior works [2, 4, 8, 20, 23]. Below we describe how an attacker can modify the metadata.

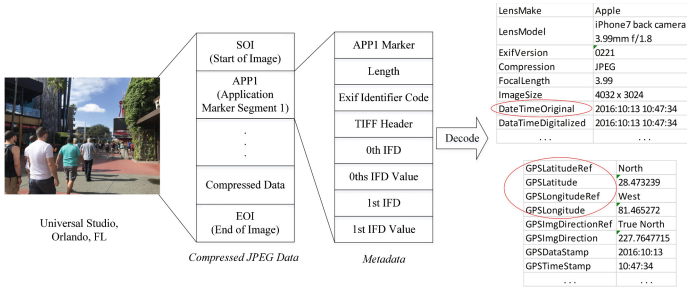


Fig. 2. The basic structure of JPEG compressed image files.

An image file contains not only the image itself but also the metadata that describes who, when, where, and how an image was taken [21, 22]. Exchangeable image file format (Exif) is a popular standard that specifies the formats of images. The specification uses the existing file formats (e.g., JPEG) with the addition of specific metadata tags. Figure 2 shows the basic structure of JPEG compressed image files [16], and the application marker segment I (APP1) contains contextual information of images, e.g., the capture time, the image size, compression format, and details of cameras (focal length, camera maker) [16], etc. In particular, `DateTimeOriginal` records the capture time. `GPSLatitude` and `GPSLongitude` contain the GPS location (i.e., latitude and longitude) of where the image was taken. `GPSImgDirection` represents the direction measured by the magnetometer (i.e., the direction in which the camera faces). Modifying the capture time and the GPS information enclosed in metadata can be easily accomplished by using metadata editing tools such as ExifTool [12]. For example, the photo shown in Fig. 2 was taken in Orlando, FL, on 13th October 2016, at 10:47 a.m. An attacker can claim that the photo was taken in May 2016 in Los Angeles by changing `DateTimeOriginal` and other related GPS fields.

2.2 Overview of AYL

Our goal is to validate whether the claimed capture time and location are true. The capture time indicates the date and time when the photo was taken, and the capture location reveals where it was taken.

Basic Idea. Although an attacker can modify the metadata and claim that a photo was taken at time X and location Y , she won’t be able to change the “time” and “location” information that is embedded in the photo. Thus, our framework works as follows. On one hand, we utilize the contents in outdoor images—vertical objects and resulting shadows—to extract the sun position that reflects when and where the image was taken. On the other hand, we utilize the metadata information—the claimed capture time and location—to obtain a second estimation of the sun position by applying astronomical algorithms. If these two estimations are close enough, we consider the capture time and location to be true with a high probability. Otherwise, they are considered to be falsified.

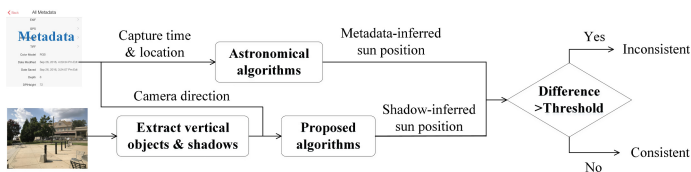


Fig. 3. The work flow of the proposed AYL framework.

Assumption. Without loss of generality, we assume that photos are taken using smartphone’s rear cameras and the smartphone is held in such way that the camera looks at the front horizontally or vertically, and it is perpendicular to the ground. We further assume that the photographer stands on ground and the ground that interested shadows lie on is approximated to water level. Finally we assume that at least one vertical object and its shadow can be seen in the image. The objects can be human beings, road signs, lampposts, tree trunks and so on.

Workflow. Figure 3 shows the work flow of proposed approach. For convenience of description, we use the term *shadow-inferred sun position* to refer to the sun position estimated from shadows in the image, and use the term *metadata-inferred sun position* to refer to the sun position calculated from claimed capture time and location. In this paper, *capture time* denotes the date of year and the time of day unless otherwise indicated.

2.3 Research Challenges

Shadow-Inferred Sun Position. The first challenge is how to obtain sun position from a single image. Although shadows can be viewed in images, we still need to know the length ratio of objects and their shadows and the orientation of shadows to determine the sun position. However, the relative position of two objects in real world is no longer preserved when they are projected to a 2-d image. How to measure the actual length ratio and angles can be a challenging problem. Although single view reconstruction has been extensively studied, there is no generalized way to recover the relative positions of objects from one single image. To address above challenges, algorithms based on projective geometry are proposed in Sect. 4.

Validation. Once the *shadow-inferred sun position* is obtained, the next challenge would be how to validate that the capture time and location are true. To estimate the true capture time and location directly from the *shadow-inferred sun position* is difficult since a specific sun position can be viewed at various places and times. Conversely, the claimed capture time and location can determine a unique value of sun position that should be close enough to the *shadow-inferred sun position*. Then, we convert previous problem into a new problem: how to determine the two estimations—the *shadow-inferred sun position* and the *metadata-inferred sun position*—are close enough indicating the

same sun position. Appropriate thresholds need to be selected to solve this problem. The sun moves across the sky at a varying speed. The changes of sun position with respect to time and location on the earth are not constant, which further complicates the selection of thresholds. We will discuss this problem in Sect. 5 and our experimental results are presented in Sect. 6.

3 Background

We discuss the basics on how the sun changes its position in the section, which serves as the foundation of our algorithm.

3.1 Sun Position Definition

The position of the sun in the sky is defined by an azimuth angle and an altitude angle. An azimuth angle describes the direction of the sun, whereas an altitude angle defines the height of the sun [24]. As shown in Fig. 4, the sun azimuth angle A is measured clockwise in the horizontal plane, from the north to the direction of the sun. Its value varies from 0° (north) through 90° (east), 180° (south), 270° (west), and up to 360° (north again). The altitude angle h is measured from the horizontal to the sun and it thus ranges from -90° (at the nadir) through 0° (on the horizon), up to 90° (at the zenith). For instance, when the sun crosses the meridian, its azimuth is 180° and altitude is at its largest value in a day.

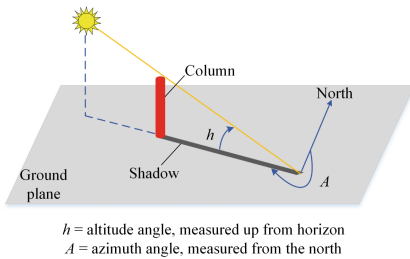


Fig. 4. An illustration of the altitude and azimuth angles of the sun.

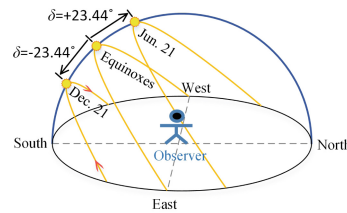


Fig. 5. The path of the sun across the sky as observed on various dates in the northern hemisphere.

3.2 How Does the Sun Move

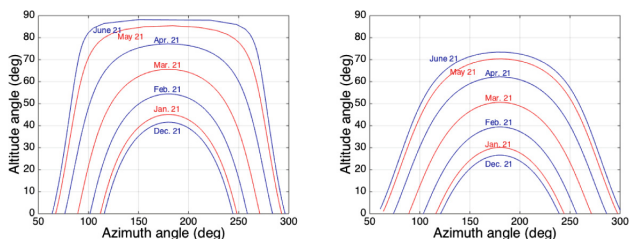
Observed from any location on the earth, the sun moves continuously across the sky throughout days and years. The relative position change is mainly caused by two types of motions of the earth: the rotation around its axis, and the revolution around the sun [24]. It takes about 24 h for the earth to finish one rotation around the earth's axis and about 365 days to complete one revolution around the sun. For an observer on the earth, the first motion contributes to the

alternation of day and night, and the second motion leads to the alternation of seasons.

Daily Sun Path. Because of the earth’s daily rotation, the sun appears to move along with the celestial sphere every day. It makes a 360° journey around the celestial sphere every 24 h. To an observer on the earth, the sun rises somewhere along the eastern horizon, and goes up to the highest point (zenith) around the noon, then goes down until it sets along the western horizon. Figure 5 shows three of the sun’s daily paths viewed on the earth. Accordingly, the cast shadows of any objects move oppositely from somewhere along the west to somewhere along the east. The shadows’ lengths vary with the sun’s altitude angle. They become shorter and shorter since sunrise and reach the shortest when the sun is at its zenith. Then they become longer over time till sunset. Thus, the shadow that a camera takes at the same day and location but different times of the day will be totally different.

Yearly Sun Path. The sun’s daily path across the sky also changes throughout the year. This is because the earth does not rotate on a stationary axis and the tilt in the axis varies each day with respect to the earth’s orbit plane. To an observer on the earth, the sun looks higher in the summer than it looks in the winter at the same time in the day. As shown in Fig. 5, the sun follows different circles at different days in one year: most northerly on June 21st and most southerly on December 21st. The sun’s motion along the north-south axis over a year is known as the declination of the sun, denoted by δ . Thus, the sun position inferred from photos taken at the same location and time but different days in a year will be different due to the sun’s declination.

Sun Path at Different Latitudes. As the sun travels across the sky, the observed altitude angle varies based on the latitude of the observer. The further north or south we go from the equator, the lower the sun’s altitude becomes. Figure 6 shows the sun’s altitude angle versus the azimuth angle observed at 25° north latitude and 40° north latitude respectively. The sun’s altitude angle observed at 25° north latitude is higher than the altitude angle observed at 40° north latitude at the same time. Thus, the sun position inferred from photos taken at the same time but different latitudes will be different.



(a) at 25° north latitude in the U.S. (b) at 40° north latitude in the U.S.

Fig. 6. The same path of the sun observed at two latitudes.

4 Shadow-Inferred Sun Position

The framework AYL uses both the azimuth angle and altitude angle to determine the position of the sun in the sky. As shown in Fig. 4, the sun’s altitude angle equals the angle between the shadow and the sun ray, and the sun’s azimuth angle equals the angle measured across the shadow point of the top of the column, clockwise from the north to the direction of the shadow. In this section, we provide algorithms to estimate the altitude and azimuth angles of the sun from shadows in a photo. We study two scenarios and two corresponding algorithms to estimate the altitude angle. We also design an algorithm to measure the azimuth angle. For both algorithms, their sensitivities are analyzed.

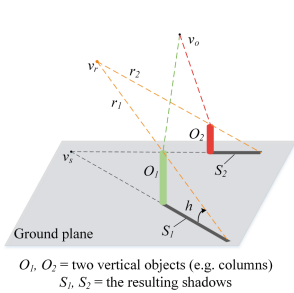


Fig. 7. Estimate the sun’s altitude angle with two shadows.

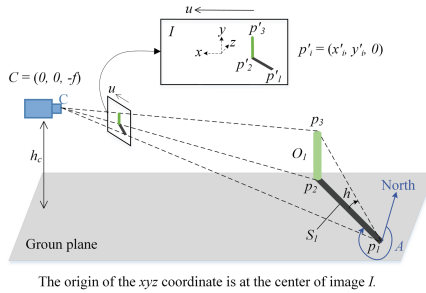


Fig. 8. Estimate the sun’s altitude and azimuth angle with one shadow.

4.1 Estimate Altitude

We consider two scenarios for estimating the sun’s altitude angles: (a) photos that contain two vertical objects and their shadows, and (b) photos that contain only one vertical object and its shadow. Vertical objects refer to the ones that are perpendicular to the ground plane.

Two-Shadow Estimation. Figure 7 illustrates the first scenario, where two objects O_1 and O_2 cast shadows S_1 and S_2 on the ground plane, respectively. The sun’s altitude angle h is the angle between the shadow and the sun ray. From the graphical perspective, a set of parallel lines in space intersect at one point when they are projected onto a 2-d image. This point is called vanishing point. In Fig. 7, shadows S_1 and S_2 of two vertical objects are parallel in space, and they intersect at vanishing point v_s on the ground plane. Since the sun is far away from the earth, the sun rays r_1 and r_2 can be considered to be parallel and intersect at v_r . The sun’s altitude angle h can be calculated according to the following formula [11]:

$$h = \arccos\left(\frac{\mathbf{v}_r^T \omega \mathbf{v}_s}{\sqrt{\mathbf{v}_r^T \omega \mathbf{v}_r} \sqrt{\mathbf{v}_s^T \omega \mathbf{v}_s}}\right), \tag{1}$$

Algorithm 1. Estimating the altitude angle h

Input: I : an image, f : the camera's focal length

Output: h

- 1: $G \leftarrow$ find the equation of the ground plane;
 - 2: find the equations of lines $\{p'_1p_1, p'_2p_2, p'_3p_3\}$;
 - 3: $\{(x_1, y_1, z_1), (x_2, y_2, z_2)\} \leftarrow$ compute the coordinates of the points $\{p_1, p_2\}$ by solving a set of equations G and $p'_i p_i$ accordingly;
 - 4: find the equation of line p_2p_3
 - 5: $(x_3, y_3, z_3) \leftarrow$ compute the coordinates of p_3 by solving a set of equations p_2p_3 and p'_3p_3 ;
 - 6: $\{\overrightarrow{p_1p_2}, \overrightarrow{p_1p_3}\} \leftarrow \{(x_2 - x_1, y_2 - y_1, z_2 - z_1), (x_3 - x_1, y_3 - y_1, z_3 - z_1)\}$
 - 7: $h \leftarrow$ compute the angle between $\overrightarrow{p_1p_2}$ and $\overrightarrow{p_1p_3}$ using Eq. 9
 - 8: **return** h
-

where ω is called the image of the absolute conic and given by the expression [11, 27]:

$$\omega \sim \begin{bmatrix} 1 & 0 & -u_0 \\ 0 & 1 & -v_0 \\ -u_0 & -v_0 & f^2 + u_0^2 + v_0^2 \end{bmatrix}. \quad (2)$$

This expression assumes that the camera has zero skew, the intersection of the optical axis and the image plane is at the center of the image, and the pixels are square. Such assumptions are true for current camera technologies [11, 27]. In Eq. 2, (u_0, v_0) denotes the coordinates of the center point of the image, and f denotes the camera's focal length. f is either included in the metadata of the image or can be calculated by the following constraint on ω with respect to f :

$$\mathbf{v}_s^T \omega \mathbf{v}_o = 0, \quad (3)$$

where \mathbf{v}_o is the vanishing points of the two vertical objects O_1 and O_2 . When the objects and their shadows are at perpendicular directions, \mathbf{v}_o and \mathbf{v}_s will satisfy Eq. 3 [11]. Once we have the coordinates of \mathbf{v}_o and \mathbf{v}_s , we can obtain f by solving Eq. 3.

One-Shadow Estimation. In this scenario, only one vertical object and its shadow are visible in the image. Figure 8 illustrates this scenario where C denotes the camera and I is the image. We assume that the image plane is perpendicular to the ground plane and the direction \vec{u} is parallel to the ground plane. So the angle between the image plane and the ground plane is 90° . Let's denote the image plane to be $z = 0$, and the coordinate frame is shown in Fig. 8. The center of the image is the origin point $(0, 0, 0)$.

Algorithm 1 describes the steps to measure the altitude angle h given a vertical object and its shadow. Firstly, to find the equation of the ground plane G , we define the distance between the camera and the ground plane to be h_c . As we know G is perpendicular to the XY plane of the coordinate system, the equation of G can be written as:

$$y = -h_c. \quad (4)$$

Next, we compute the equations of the lines p'_1p_1 , p'_2p_2 and p'_3p_3 . Since the line p'_ip_i passes through the point $C(0, 0, f)$ and the point p'_i whose coordinates can be obtained from the image, it can be described by the two points as:

$$\frac{x}{x'_i} = \frac{y}{y'_i} = \frac{z + f}{f}, \quad (5)$$

where $(x'_i, y'_i, 0)$ are the coordinates of p'_i for $i = 1, 2, 3$.

Lines p'_1p_1 and p'_2p_2 intersect with plane G at points p_1 and p_2 respectively. By solving the Eqs. 4 and 5, the coordinates of p_1 and p_2 can be computed as follows:

$$p_i = (x'_it_i, -h_c, f(t_i - 1)), \quad (6)$$

where $t_i = -\frac{h_c}{y'_i}$ for $i = 1, 2$. Then we have vector $\overrightarrow{p_1p_2} = (x'_2t_2 - x'_1t_1, 0, f(t_2 - t_1))$.

Now, we determine the coordinates of p_3 which is the intersection point of lines p'_3p_3 and p_2p_3 . The equation of line p_2p_3 is given by:

$$x = x'_2t_2, \quad z = f(t_2 - 1). \quad (7)$$

By solving the equations of p'_3p_3 and p_2p_3 , we can obtain the coordinates of the point p_3 :

$$p_3 = (x'_2t_2, y'_3t_2, f(t_2 - 1)). \quad (8)$$

Using the coordinates of p_2 and p_3 , we have vector $\overrightarrow{p_1p_3} = (x'_2t_2 - x'_1t_1, h_c + y'_3t_2, f(t_2 - t_1))$. The angle between $\overrightarrow{p_1p_3}$ and $\overrightarrow{p_1p_2}$ is the altitude angle and can be computed as follows:

$$\begin{aligned} h &= \arccos \frac{(\overrightarrow{p_1p_3})^T \overrightarrow{p_1p_2}}{\sqrt{(\overrightarrow{p_1p_3})^T \overrightarrow{p_1p_3}} \sqrt{(\overrightarrow{p_1p_2})^T \overrightarrow{p_1p_2}}}, \\ &= \arccos \frac{m}{\sqrt{m + (y'_3/y'_2 - 1)^2 \sqrt{m}}}. \end{aligned} \quad (9)$$

where the intermediate variable $m = (\frac{x'_2}{y'_2} - \frac{x'_1}{y'_1})^2 + f^2(\frac{1}{y'_2} - \frac{1}{y'_1})^2$.

4.2 Estimate Azimuth

To estimate the sun's azimuth angle A from one shadow in an image, we design the following algorithm. The scenario is illustrated in Fig. 8. In particular, the point p_3 is not necessary to be visible for estimating the azimuth angle. Let C be the camera and the unit vector $\vec{u} = (1, 0, 0)$. The true north N is set to be the reference direction in our algorithm. The orientation of \vec{u} with respect to N can be obtained by subtracting 90° from the image direction which is included in the metadata of the image.

The sun azimuth angle A equals the angle measured clockwise around point p_1 from due north to the shadow. We calculate A as follows:

$$A = \angle(N, \vec{u}) + \angle(\vec{u}, \overrightarrow{p_1p_2}), \quad (10)$$

where $\angle(\vec{u}, \overrightarrow{p_1 p_2})$ denotes the angle measured clockwise from \vec{u} to $\overrightarrow{p_1 p_2}$, and $\angle(N, \vec{u})$ is the angle measured clockwise from N to \vec{u} , which is the orientation of \vec{u} . $\angle(\vec{u}, \overrightarrow{p_1 p_2})$ is the only unknown variable in Eq. 10.

Next, we define the angle between \vec{u} and $\overrightarrow{p_1 p_2}$ to be α . $\angle(\vec{u}, \overrightarrow{p_1 p_2})$ equals α if it is an acute angle. Otherwise, $\angle(\vec{u}, \overrightarrow{p_1 p_2})$ is equal to $(360^\circ - \alpha)$. The angle α can be calculated as:

$$\alpha = \arccos \frac{\vec{u}^T \overrightarrow{p_1 p_2}}{\sqrt{\vec{u}^T \vec{u}} \sqrt{\overrightarrow{p_1 p_2}^T \overrightarrow{p_1 p_2}}}, \quad (11)$$

where $\overrightarrow{p_1 p_2}$ has been calculated in Algorithm 1: $\overrightarrow{p_1 p_2} = (x'_2 t_2 - x'_1 t_1, 0, f(t_2 - t_1))$ and $\vec{u} = (1, 0, 0)$. Then, we replace \vec{u} and $\overrightarrow{p_1 p_2}$ in Eq. 11 and compute it as:

$$\alpha = \arccos\left(\frac{\left(\frac{x'_1}{y'_1} - \frac{x'_2}{y'_2}\right)}{\sqrt{\left(\frac{x'_1}{y'_1} - \frac{x'_2}{y'_2}\right)^2 + f^2\left(\frac{1}{y'_1} - \frac{1}{y'_2}\right)^2}}\right). \quad (12)$$

4.3 Sensitivity Analysis

In this section, we quantify the estimation errors in the computing of the altitude angles and azimuth angles.

Errors of the Altitude Angle Inferred from Two Shadows. The estimation errors of altitude angles stem from the following factors: camera distortion and the detection errors of the objects and shadows. For a well designed camera, the systematic errors (e.g. camera distortion) are constant and can be calibrated if necessary. The detection errors of the shadows and objects in the image can be modeled as random variables. Consequently, the detection errors will result in random errors in the calculation of v_r and v_s . Without loss of generality, we consider the errors of v_r and v_s to be linear to the detection errors and define them to be Δv_r and Δv_s , respectively.

From the graphical perspective, the sun altitude angle h derived from vanishing points v_r and v_s has the geometric meaning as described in Fig. 9. Let C be the camera. The lines Cv_r and Cv_s are parallel to the shadow and the sun ray respectively. h represents the sun altitude angle and equals the angle formed by v_r , C and v_s . The error range of each vanishing point is a circle centered at

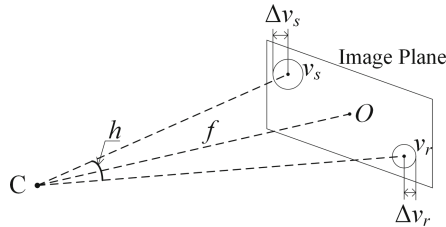


Fig. 9. Errors in the estimated altitude angle with two shadows.

the vanishing point with a radius of the maximum random error. Since Δv_r and Δv_s are small enough compared to the length of $|Cv_r|$ and $|Cv_s|$, they can be considered as two arcs with the center at C . In the worst case, the error Δh of the altitude angle can be calculated as below:

$$\Delta h = \left(\frac{\Delta v_r}{|Cv_r|} + \frac{\Delta v_s}{|Cv_s|} \right) \frac{180^\circ}{\pi}, \quad (13)$$

where $|Cv_r|$ and $|Cv_s|$ are the lengths between the camera and the two corresponding vanishing points: v_r and v_s . Thus, the error Δh depends on the random errors Δv_r and Δv_s .

Errors of the Altitude Angle Inferred from One Shadow. The sources of random errors in estimating the altitude angle from one shadow include the slope of the ground and the detection errors of interested object and its shadow. If the ground where the shadow located is not flat and has an error of ΔG with respect to the horizontal plane, ΔG will propagate as the altitude angle is estimated. In addition, the detection errors of the vertical object and its shadow can cause estimation errors. The detection errors can affect the angle estimation depending on the distances from the camera to the object and its shadow. The farther the distance, the larger the uncertainty of the estimated altitude angle. The errors in the altitude angle can be linear to the detection errors.

Errors of the Azimuth Angle. The sources of random errors in estimating the azimuth angle include the camera's orientation errors and the detection errors of shadows. To understand how a camera's orientation affects the estimation error of the angle between \vec{u} and the shadow S_1 , we define θ to be the angle between the image plane and the horizontal ground plane, and γ to be the angle between the camera and the horizontal plane. Assume $\theta = 90^\circ$ and $\gamma = 0^\circ$. And the estimated camera orientation is $\theta = 90^\circ + \Delta\theta$ and $\gamma = 0^\circ + \Delta\gamma$, where $\Delta\theta$ and $\Delta\gamma$ are random errors.

Figure 10 shows the impact of $\Delta\theta$ and $\Delta\gamma$ on the estimated direction of the shadow. First, the error $\Delta\theta$ will be propagated as we estimate the ground plane according to the camera's orientation. And due to this error, the estimated shadow direction will deviate from the true direction of the shadow. The deviation will be $\Delta\theta$ in the worst case. Second, the error $\Delta\gamma$ will also be propagated to the estimated ground plane. And this error will lead to a deviation in the estimated shadow direction as well, which is $\Delta\gamma$ in the worst case. In summary, the estimated shadow direction deviates from its true direction at most $\Delta\theta + \Delta\gamma$, which can produce $\Delta\theta + \Delta\gamma$ error in the estimated azimuth angle in the worst case.

In summary, we find three main sources of the errors: the detection errors of the objects and their shadows, the ground slope, and the camera's orientation errors. In general, the estimation errors of the sun position are linear to the three types of errors. The detection errors in our algorithms can be reduced by choosing the objects and shadows that are clear enough and using effective image detection algorithms. The errors caused by the slope of the ground will not be

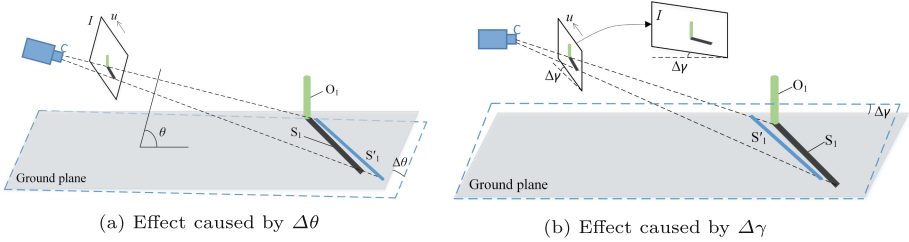


Fig. 10. Effects caused by random errors in camera’s orientation

greater than the slope angle and can be reduced greatly by measuring this angle. In addition, the camera’s orientation errors can be reduced using inertial sensors to obtain the camera orientation.

5 Metadata-Inferred Sun Position and Validation

In this section, we describe the process to validate the consistency of a photo’s capture time and location. The key idea is the following: we calculate the sun position using the capture time and location in the metadata of images. If the capture time and location are true, the sun position will match the one we estimated from shadows.

5.1 Metadata-Inferred Sun Position

As mentioned in Sect. 3, the position of the sun depends on the time of day, the date and the location of the observer. Its movement across the sky obeys the rules that have been studied in astronomy. In this section, we discuss the astronomical algorithms that are used to calculate *metadata-inferred sun position*, given the time and location.

We refer the time of day as the local time based on the standard time offsets of Coordinated Universal Time (UTC). However, the local standard time doesn’t provide an intuitive connection with the sun position. In astronomy, the solar time is often used to discuss the sun position. It works because the sun finishes a 360° rotation around the celestial sphere every 24 h. The completed journey is divided into 24 h, and one solar hour means that the sun travels a 15° arc [19]. The instant when the sun is due south in the sky or the shadow points to exactly north is called solar noon, which is 12:00 for solar time. Every 15° arc the sun travels, one hour is added to 12:00 under the 24-h clock system, and the angle distance that the sun passes on the celestial sphere is defined as the hour angle H [19]. It is measured from the sun’s solar noon position, and ranges from 0° to +180° westwards and from 0° to -180° eastwards. The conversion between the local standard time t_l to the solar time t_s is as follows [13, 24]:

$$t_s = t_l + ET + \frac{4 \text{ min}}{\text{deg}} (\lambda_{std} - \lambda_l), \tag{14}$$

where λ_l denotes the local longitude, and λ_{std} is the local longitude of standard time meridian, and ET stands for the equation of time, which describes the difference of the true solar time and the mean solar time [13]. The sun's hour angle is calculated as follows:

$$H = 15^\circ(t_s - 12). \quad (15)$$

Using the observer's local horizon as a reference plane, the azimuth and altitude angles of the sun can be calculated as follows [24]:

$$\tan(A) = \frac{\sin H}{\sin \varphi \cos H - \cos \varphi \tan \delta}, \quad (16)$$

$$\sin(h) = \sin \delta \sin \varphi + \cos \varphi \cos \delta \cos H, \quad (17)$$

where φ is the latitude of the observer's location, and δ is the sun's declination angle and it can be calculated as below [15, 24]:

$$\delta = -23.44^\circ \cos\left(\frac{360^\circ(N + 10)}{365^\circ}\right), \quad (18)$$

where N is the number of days since January 1st. Note that the azimuth angle A calculated in Eq. 16 uses south as a reference. We can derive the azimuth angle according to its definition in Sect. 3.

5.2 Consistency Validation

Once obtaining the *shadow-inferred sun position* and *metadata-inferred sun position*, we check the difference between these two estimations by comparing their altitude angles and azimuth angles respectively. However, since there exists random and systemic errors in the *shadow-inferred sun position*, the estimation may not equal the "true" sun position. Thus, we have to select a threshold that is large enough to tolerate the errors yet small enough to detect the inconsistency between the *shadow-inferred sun position* and *metadata-inferred sun position*. Intuitively, the closer these two sun positions are to each other, the more likely the capture time and location are true.

We define the altitude angles of *shadow-inferred sun position* and *metadata-inferred sun position* to be h_s and h_m respectively, and the corresponding azimuth angles to be A_s and A_m . Then the distance of the two altitude angles is $d_h = |h_s - h_m|$, and the distance of the two azimuth angles is computed as $d_A = |A_s - A_m|$. The likelihood of the consistency is inversely proportional to d_h and d_A . However, the effects on d_h and d_A caused by fake capture time and/or location are different. For example, modifying the capture time from 12:00 p.m. to 13:00 p.m. may lead to 10° in d_A but only 2° in d_h . So two different thresholds for d_h and d_A have to be selected. The capture time and location are considered to be true only when both d_h and d_A are within the thresholds. Besides, the sun position can be described by a pair of azimuth angle and altitude angle: (A, h) .

We can also use the sun position distance that is computed as $d_p = \sqrt{d_A^2 + d_h^2}$ to distinguish the two estimations of the sun position. Our goal is to choose appropriate variables and thresholds that can increase the probability of correct validation for inconsistent images and decrease the probability of false validation for consistent images. Section 6 details the selection of thresholds in the validation experiment.

6 Evaluation

This section presents the results of our experiments. To evaluate the performance of the sun position estimation algorithms, we conducted an experiment on November 8, 2016 in the U.S. and collected 60 photos. To validate the effectiveness of the framework AYL, we gathered 124 photos in China and the U.S in the span of four months, and examined whether we can detect the modifications of capture time, date and location.

6.1 Sun Position Estimation

To evaluate the accuracy of our sun position estimation algorithms, we collected 60 photos using the rear camera of an iPhone 7 from 9:30 a.m. to 14:30 p.m. at an interval of about 5 min on November 8, 2016 in Columbia, SC. As shown in Fig. 11(a) we set up the experiment in a relatively ideal situation: we place two columns (the red one and the grey one) on the ground vertically, and fixed the iPhone 7 on another vertical stick to take photos of these two columns and their shadows. Figure 11(b) shows the estimated altitude angles by applying the **two-shadow estimation** and **one-shadow estimation** algorithms to the photos. The ground truth sun positions are calculated using the astronomical algorithms in Sect. 5.1 according to the real time, latitude and longitude. The ground truth altitude angles are labeled with red and denoted as “Altitude”. The other two

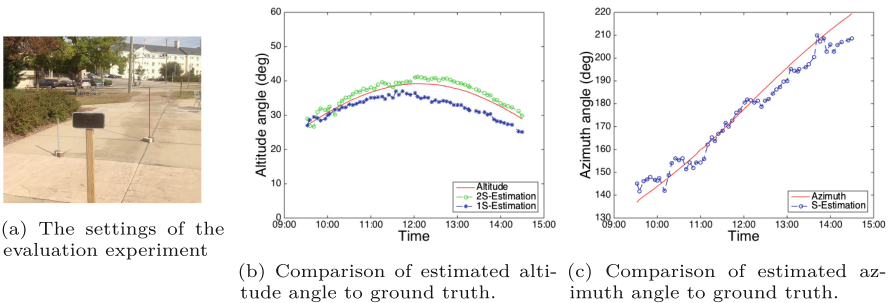


Fig. 11. The experiment setting is shown in (a). And (b) presents the comparison of estimated altitude angle to ground truth altitude angle. (c) shows the comparison of estimated azimuth angle to ground truth azimuth angle.

curves in Fig. 11(b) represent the estimated altitude angles inferred from shadows. “2S-Estimation” is obtained by applying the **two-shadow estimation** algorithm, while “1S-Estimation” is plotted by applying the **one-shadow estimation** algorithm. We find that the average error in “2S-Estimation” is 1.43° , while it is 2.98° in “1S-Estimation”. Figure 11(c) presents the estimated azimuth angles versus the ground truth azimuth angles. The curve in red is plotted using the ground truth azimuth angles, while the curve in blue is plotted using the data of estimated azimuth angles. The average error is approximately 4.3° .

The estimation errors of sun positions are mainly contributed by three factors. First, due to the ground slope and the camera’s orientation, the image plane may not be precisely perpendicular to the ground plane, which causes errors. The second type of errors is random one that is introduced when extracting objects and shadows from the photos. Finally, errors can be created by the measurement drift of the compass over time. Due to the nature of the two algorithms, these types of errors will have different levels of impact on them. Figure 11(b) indicates that the **two-shadow estimation** algorithm outperforms the **one-shadow estimation** algorithm. It is partly because the **one-shadow estimation** algorithm is more sensitive to the ground slope. We believe that given the measurement of the slope, we shall be able to reduce the error. In summary, the algorithms in Sect. 4 are able to infer the sun position, either from two vertical objects and their shadows or from one object and its shadow.

6.2 Consistency Validation

To evaluate the performance of AYL and to understand threshold selection, we conducted a set of experiments.

Dataset. The data in this experiment was captured at 15 cities around the USA and China since September 2016. Our dataset consists of 124 photographs taken by 10 iPhones, including iPhone 5s, 6, 6 plus, 6s, 6s plus and 7. 61 out of the 124 photos were taken in China. Each photo encloses the metadata that includes the real capture time and location. 72 out of the 124 photos contain at least two vertical objects and their shadows, while 52 photos only contain one vertical object and its shadow. Our dataset mainly contains three types of vertical objects: standing people, poles (e.g. road signs, lampposts) and tree trunks. We chose these objects because they are common in reality and are mostly vertical to the ground. Our experimental results confirm that our algorithms work well on these objects. We refer to the true metadata of the 124 photos as the *positive samples*.

We generate the attack data by falsifying the metadata of the 124 photos. Note that multiple types of metadata may result in the same effect. For instance, modifying longitude one degree more to the west has the same effect on the sun position as changing the local time forward by four minutes. Thus, falsifying either longitude or the local time is equivalent. To simplify the analysis yet without loss of generality, we focus on three types of attacks that modify the following metadata:

- The falsified time of day, and true date and location.
- The falsified date, and true time and location.
- The falsified latitude of location, and true time and date.

We refer to the attack metadata as the *negative samples*. We have 124 negative samples for each type of attack metadata. The “fake” times of day are randomly generated in the range from 8:00 a.m. to 17:00 p.m. when the sun is likely to be seen. The “fake” dates are randomly generated from the range within one year. The “fake” latitudes of location are randomly generated in the range of 25° and 50° of the Northern Hemisphere where most of the U.S. and China locate. Here we didn’t consider the attack data with falsified longitude. Because the result produced by only falsifying longitude can be equivalent to the result caused by falsifying the time of day accordingly.

Metric. We use ROC curves to evaluate the performance of AYL by varying thresholds for our system. An ROC curve represents Receiver Operating Characteristic curve and is created by plotting true positive rate (TPR) against false positive rate (FPR), as the threshold varies [6]. The true positive rate and false positive rate are defined as below.

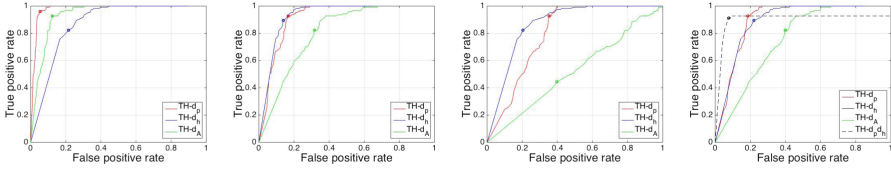
$$TPR = \frac{\# \text{ of true positives}}{\# \text{ of (true positives + false negatives)}} = \frac{\# \text{ of true positives}}{\# \text{ of positives}}$$

$$FPR = \frac{\# \text{ of false positives}}{\# \text{ of (true negatives + false positives)}} = \frac{\# \text{ of false positives}}{\# \text{ of negatives}}$$

where a true positive denotes the result that a positive sample is correctly identified as such, and a false positive is the one that a negative sample is identified as a positive sample by mistakes. The point (0, 1) on the ROC curve denotes 0 FPR and 100% TPR , which indicates an ideal system that can correctly identify all genuine photos and reject all falsified photos [25]. In our experiment, we select the optimal threshold as the one that yields the minimum distance from the corresponding point on the ROC curve to the ideal point (0, 1). Another indicator that we use to evaluate the average performance of the validation is the area under the ROC curve (AUC). The closer it is to 1, the better the average performance is [6].

Performance and Threshold Selection. Based on the framework AYL, we performed consistency validation using the three types of attack metadata. To understand how the altitude angle and azimuth angle influence the performance of the validation, we examine three distances separately: the distance of the altitude angles d_h , the distance of the azimuth angles d_A , and the distance of the sun positions d_p . Here, the sun position is defined to be (A, h) , in which A refers to the azimuth angle and h refers to the altitude angle. To decide the best distance variable which can yield the maximum AUC and the optimal threshold of the variable, we analyze the ROC curves that are plotted by varying the threshold of each type of distance.

The results are presented in the set of ROC curves shown in Fig. 12. Each ROC curve with distinct color is plotted by varying the threshold of one type



(a) The attack meta- (b) The attack meta- (c) The attack meta- (d) A collective of the
data with falsified time data with falsified data with falsified three types of attack
of day. date. latitude. metadata.

Fig. 12. ROC curves based on different distance variables and different types of attack metadata.

of the three distances. “TH – d_p ” denotes varying the threshold of the sun position distance d_p . “TH – d_h ” and “TH – d_A ” denote varying the threshold of the altitude angle distance d_h and the azimuth angle distance d_A respectively. For each type of attack metadata, the randomly generating of 124 negative samples is repeated 5 times. Each false positive rate on the ROC curve is averaged over these repeated attack metadata. Figure 12(a) indicates that the detection based on d_A slightly outperforms the one based on d_h , to detect the attacks that falsify the photo’s time of day. However, the d_h based detection achieves better performance in detecting the other types of attacks as shown in Fig. 12(b–c), especially in detecting falsified latitude. The result implies that d_h is more important in distinguishing different positions of the sun compared to d_A in general. Such a conclusion confirms with the result reported in Sect. 6.1, i.e., the average estimation error of the altitude angles is smaller than that of the azimuth angles. If only d_h is used for consistency validation, Fig. 12(d) guides us to choose the optimal threshold of d_h to be 3° and it achieves combined (TPR , FPR) values of (89.5%, 22%), which means that 89.5% of positive samples can be correctly validated but 22% of negative samples will be mistakenly identified. In addition, Fig. 12(a–c) shows that the d_p based detection achieves the best performance in detecting falsified time, and has almost the same performance as the d_h based detection in detecting the other types of attacks. Once we only use d_p for consistency validation as shown in Fig. 12(d), we choose the optimal threshold of d_p to be 9.2° , which achieves combined (TPR , FPR) values of (92.7%, 18.6%) for all attacks.

To improve the performance further, we examine both the d_p and d_h to validate the consistency of time and location. That is, a sample has to satisfy both the thresholds of d_h and d_p to be accepted by AYL. Plotting the ROC curves and finding the global optimal thresholds by varying two thresholds can be tricky. Thus, we chose the local optimal threshold for one variable and varied the other threshold to plot the ROC curve. This approach may not generate the global optimal thresholds for the two variables, but it strikes a balance between the optimum and the computational cost. We chose the threshold of d_p to be 9.2° and varied the threshold of d_h . The resulting curve illustrates an improved performance than the one of using a single threshold as shown in Fig. 12(d).

Note that we cannot plot an integral ROC curve when the threshold of d_p is fixed since the highest true positive rate will be decided by the fixed threshold, which is 92.7%. The curve “TH – $d_p d_h$ ” in Fig. 12(d) indicates that choosing the optimal threshold of d_h to be 4.8° can correctly identify 91.1% positive samples, and cannot identify 7.7% of negative samples.

Attacks Against AYL. Based on the above results, we analyze the robustness of the framework AYL when falsifying one of the three parameters—time of day, date and latitude of location, and falsifying more than one parameters. AYL cannot detect the falsifications that do not cause violations of both the thresholds of the altitude angle distance and the sun position distance. If an attacker modifies both the time and location of a photo such that the altitude angle and the sun position are within the thresholds, then the modification can fool AYL. Luckily, the motivation of falsifying the metadata of a photo is to use it for a chose event and the attacker may not be guaranteed to find such a combination.

Our framework can detect that the image shown in Fig. 1 was not taken at the claimed date and location. Although we do not have the required metadata (e.g., the time of day, the image direction and the camera orientation) and cannot estimate the azimuth angle as well as the exact sun position, we can estimate the altitude angle from the image. Given that the photo was claimed to be taken on or before January 16th in Florida, we can calculate the possible maximum altitude angle between January 1st and 16th to be 41° . Based on the image, we estimate the focal length to be 1287 pixels and the altitude angle to be 47.6° . The distance between the two estimates will be 6.6° which is larger than the threshold 4.8° in our experiments. Thus, we conclude that the date and location of this image were spoofed.

6.3 Discussions and Limitations

When estimating the altitude angle using the **one-shadow estimation** algorithm, an integrated vertical object and its shadow are required. However, vertical objects in the real world may not be absolutely vertical. By examining the scenario, we find that the direction of the sun ray is determined by a point on the object and the resulting point on the shadow. Even if the object is not exactly vertical, these two points are still able to decide the path of the sun ray and the altitude angle can be obtained from the sun ray and the shadow. Thus, we believe that our algorithm can eliminate this requirement, and AYL may not require to have the entire object in the photo if there exists a distinct point on the object.

In this paper, we assume that the camera is perpendicular to the ground and looks front horizontally or vertically when taking photos. Such assumption is used to simplify the algorithms for estimating the sun position. In fact, most smartphotnes are equipped with inertial sensors that have been widely used to estimate the orientation of the smartphone. If the sensor data is enclosed in the metadata, the device orientation can be obtained and used to determine the relationship between the device and the ground as well as the shadows.

A direction of future work is to estimate the sun position regardless of how cameras are oriented when taking photos.

7 Conclusion

We presented a new framework AYL which uses two estimations of sun position—*shadow-inferred sun position* and *metadata-inferred sun position*—to check whether the capture time and location of an outdoor image are true. Our framework exploits the relationship between the sun position in the sky and the time and location of an observer. We designed algorithms to obtain *shadow-inferred sun position* using only one vertical object and its shadow in the image. Our experiments show that the algorithms can estimate the sun position from shadows in the image with satisfactory accuracy. AYL utilizes both the altitude angle and azimuth angle for the consistency validation. The evaluation results guide us to choose the thresholds of altitude angle distance and sun position distance to be 4.8° and 9.2° respectively, which achieves combined (TPR , FPR) values of (91.1%, 7.7%) for the consistency validation. We believe that our results illustrate the potential of using sun position to validate the consistency of the capture time and location. Our work raises an open question that whether other image contents can be leveraged for validating the consistency of image’s contextual information.

References

1. Fema now has a rumor control section for misinformation. <https://twitter.com/fema/status/264800761119113216>
2. Bayram, S., Sencar, H.T., Memon, N.: An efficient and robust method for detecting copy-move forgery. In: IEEE ICASSP 2009, pp. 1053–1056. IEEE (2009)
3. Boididou, C., Papadopoulos, S., Kompatsiaris, Y., Schifferes, S., Newman, N.: Challenges of computational verification in social multimedia. In: ACM WWW 2014, pp. 743–748. ACM (2014)
4. Carvalho, T., Farid, H., Kee, E.: Exposing photo manipulation from user-guided 3D lighting analysis. In: SPIE/IS&T Electronic Imaging, p. 940902. International Society for Optics and Photonics (2015)
5. Dearden, L.: The fake refugee images that are being used to distort public opinion on asylum seekers, September 2015. <http://www.independent.co.uk/news/world/europe/the-fake-refugee-images-that-are-being-used-to-distort-public-opinion-on-asylum-seekers-10503703.html>
6. Fawcett, T.: An introduction to ROC analysis. *Pattern Recogn. Lett.* **27**(8), 861–874 (2006)
7. Ferrara, E.: Manipulation and abuse on social media by Emilio Ferrara with Chingman Au Yeung as coordinator. *ACM SIGWEB Newslett.* (Spring), Article No. 4 (2015)
8. Fridrich, J., Soukal, D., Lukas, J.: Detection of copy move forgery in digital images. In: *Digital Forensic Research Workshop*, August 2003

9. Garg, R., Varna, A.L., Wu, M.: Seeing ENF: natural time stamp for digital video via optical sensing and signal processing. In: ACM MM 2011, pp. 23–32. ACM (2011)
10. Gupta, A., Lamba, H., Kumaraguru, P., Joshi, A.: Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In: ACM WWW 2013, pp. 729–736. ACM (2013)
11. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
12. Harvey, P.: Exiftool. <http://www.sno.phy.queensu.ca/~phil/exiftool/>
13. Holbert, K.E., Srinivasan, D.: Solar energy calculations. In: Handbook Of Renewable Energy Technology, pp. 189–204 (2011)
14. Imran, M., Elbassuoni, S.M., Castillo, C., Diaz, F., Meier, P.: Extracting information nuggets from disaster-related messages in social media. In: Proceedings of ISCRAM, Baden-Baden, Germany (2013)
15. Iqbal, M.: An Introduction to Solar Radiation. Elsevier, Amsterdam (2012)
16. Japan Electronics and Information Technology Industries Association: Exchangeable image file format for digital still cameras: Exif Version 2.2, April 2002
17. Junejo, I.N., Foroosh, H.: Estimating geo-temporal location of stationary cameras using shadow trajectories. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5302, pp. 318–331. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-88682-2_25](https://doi.org/10.1007/978-3-540-88682-2_25)
18. Kakar, P., Sudha, N.: Verifying temporal data in geotagged images via sun azimuth estimation. IEEE Trans. Inf. Forensics and Secur. **7**(3), 1029–1039 (2012)
19. Karttunen, H., Kroger, P., Oja, H., Poutanen, M., Donner, K.J.: Fundamental Astronomy, 5th edn. Springer, Heidelberg (2007)
20. Kee, E., O’Brien, J.F., Farid, H.: Exposing photo manipulation from shading and shadows. ACM Trans. Graph. **33**(5), 165:1–165:21 (2014)
21. Metadata Working Group: Guidelines For Handling Image Metadata v2.0, November 2010
22. National Information Standards Organization: Understanding Metadata (2004)
23. Pan, X., Lyu, S.: Detecting image region duplication using sift features. In: IEEE ICASSP 2010, pp. 1706–1709. IEEE (2010)
24. Savoie, D.: Sundials: Design, Construction, and Use. Praxis Publishing, Chichester (2009)
25. Tian, J., Qu, C., Xu, W., Wang, S.: Kinwrite: handwriting-based authentication using kinect. In: NDSS 2013 (2013)
26. Tsai, T.H., Jhou, W.C., Cheng, W.H., Hu, M.C., Shen, I.C., Lim, T., Hua, K.L., Ghoneim, A., Hossain, M.A., Hidayati, S.C.: Photo sundial: estimating the time of capture in consumer photos. Neurocomputing **177**, 529–542 (2016)
27. Wu, L., Cao, X.: Geo-location estimation from two shadow trajectories. In: IEEE CVPR 2010, pp. 585–590 (2010)