GEORG AUGUST UNIVERSITÄT GÖTTINGEN

Fakultät für Forstwissenschaften und Waldökologie

MASTER'S THESIS

An approach to distance estimation from camera trap pictures

written by

Santiago Soto Maurer

and examined by Dr. Johannes Signer and Prof. Dr. Niko Balkenhol

in partial fulfillment of the requirement
for the degree of M.Sc. Forest and Ecosystem Sciences
in the study track Ecosystem Analysis and Modelling.

20th of June, 2024

1 Abstract

The biodiversity crises call for accurate estimates of wildlife population densities, and camera traps with corresponding estimators offer a promising tool. Methods for density estimation require estimates of distance to photographed animals. Existing distance estimation methods are either labor-intensive or inaccurate. To address this gap, I present a novel, user-friendly application requiring only two reference objects within a camera trap image. This method uses the principles of similar triangles and geometric perspective to estimate width and position of hypothetical objects on image, forming a basis for the semi-automatic generation of a depth and lateral distance grid. The grid can be used to estimate distances in the image. Initial tests at a single camera setting demonstrate good overall accuracy with a mean error of -0.17 meters. Error expectedly increased with distance (ratio of error to measurement: ± 0.15), and field of view position interacted with distance category to influence error. The largest errors were for objects on the edge of the image and 10 meter distance. While these results are promising, further evaluation is necessary to assess the impact of uneven terrain and placement of reference objects.

1.0.1 Zusammenfassung

Die Biodiversitätskrisen erfordern genaue Schätzungen der Populationsdichte von Wildtieren, dafür können Kamerafallen und entsprechende Schätzungsmethoden als nützliche Instrumente fungieren. Methoden zur Dichteschätzung erfordern Informationen zur Entfernung von fotografierten Tieren. Bestehende Entfernungsschätzungsmethoden sind entweder arbeitsintensiv oder ungenau. Um diese Lücke zu schließen, stelle ich eine neuartige, benutzerfreundliche Anwendung vor, die nur zwei Referenzobjekte innerhalb eines Kamerafallenbildes benötigt. Diese Methode nutzt die Prinzipien der Ähnlichkeitssätze bei Dreiecken und der projektiven Geometrie, um die Breite und Position hypothetischer Objekte auf dem Bild zu schätzen und bildet die Grundlage für ein halbautomatisch gezeichnetes Tiefen- und Seitenabstandsgitter. Das Gitter kann dann zur Abschätzung von Entfernungen im Bild verwendet werden. Erste Tests in einer einzigen Landschaft zeigen eine gute Gesamtgenauigkeit mit einem mittleren Fehler von -0,17 Metern. Der Fehler nahm erwartungsgemäß mit der Entfernung zu (Verhältnis von Fehler zu Messung: $\pm 0{,}15$), und die Position des Sichtfelds wirkte sich in Wechselwirkung mit der Entfernungskategorie auf den Fehler aus. Die größten Fehler traten bei Objekten am Rande des Bildes und in einer Entfernung von 10 Metern auf. Diese Ergebnisse sind zwar vielversprechend, doch sind weitere Untersuchungen erforderlich,

um die Auswirkungen von unebenem Gelände und der Platzierung von Referenzobjekten zu bewerten.

1.0.2 Resumen

Las crisis de biodiversidad exigen estimaciones precisas de las densidades de población de la fauna salvaje, y las cámaras trampa con sus correspondientes estimadores ofrecen una herramienta prometedora. Los métodos de estimación de densidad requieren conocimiento de la distancia a los animales fotografiados. Los métodos existentes de estimación de la distancia requieren mucho trabajo o son inexactos. Para compensar este problema, presento una nueva aplicación que es fácil de usar y que sólo requiere dos objetos de referencia dentro de una imagen de cámara trampa. Este método utiliza los principios de los semejanza de triángulos y la geometría proyective para estimar la anchura y la posición de objetos hipotéticos en la imagen, formando una base para la generación semiautomática de una cuadrícula de distancias laterales y de profundidad. La cuadrícula puede utilizarse para estimar distancias en la imagen. Pruebas iniciales en un sólo lugar demuestran una buena precisión general, con un error medio de -0,17 metros. El error aumentó previsiblemente con la distancia (relación entre el error y la medición: ±0,15), y la posición del campo de visión interactuó con la categoría de distancia para influir en el error. Los mayores errores se produjeron en los objetos situados en el borde de la imagen y a 10 metros de distancia. Aunque estos resultados son prometedores, es necesario realizar más evaluaciones para valorar el impacto de las irregularidades del terreno y la colocación de los objetos de referencia.

Table of Contents

2 Introduction

The current biodiversity crisis (Ceballos et al. 2015; Díaz et al. 2019) and alarming declines in wildlife populations (McRae et al. 2022) coupled with growth of the global economy (Brand and Wissen 2013; Moore 2016), underscore the critical importance of species monitoring. Population density estimates lie at the heart of understanding population dynamics and are part of knowing species' risk of extinction (Cain et al. 2011) or threat for ecosystems (*e.g.* Valente et al. 2020). Accurately estimating population densities is thus essential for practitioners and researchers in ecology and conservation.

Estimating wildlife population densities encompasses a diverse range of methods. Traditional approaches often involve labor-intensive techniques like transect (Thompson 1994) or point surveys (Kissling and Garton 2006), or the capture and recapture of individual animals (Otis et al. 1978; Wilson and Anderson 1985). With the declining cost of camera trap technology and increase of methods for density estimation with camera trap data, camera traps have turned to be a non-invasive, cost-effective alternative (Foster and Harmsen 2012) specially for regions where labor costs exceed prices of camera equipment. For species with identifiable characteristics in camera trap photos, capture-recapture methods can be employed (Griffiths and Van Schaik 1993; Karanth and Nichols 1998). However, for species on which individual identification from camera trap data is not possible, several methods for unmarked density estimation have been developed (Gilbert et al. 2021). These methods often rely on information about the distance between captured animals and the camera (Gilbert et al. 2021; Hay 2024).

Several unmarked population density estimation methods have emerged over the last two decades: camera trap distance sampling (CTDS; Howe et al. (2017)), random encounter model (REM; Rowcliffe et al. (2008)), Random Encounter and Staying Time (REST; Nakashima et al. (2018)), and Instantaneous Sampling (IS; Moeller et al. (2018)).

CTDS, a camera trap adaptation of distance sampling (Thompson 1994), specifically uses video footage (Howe et al. 2017) but has been expanded for using still images (Kühl et al. 2023). It has been shown to be a reliable alternative to distance sampling (Corlatti et al. 2020). CTDS relies on the assumption that detection probability decreases with increasing distance from the camera. By utilizing precise distances to detected animals, CTDS can fit a detection function, estimate missed detections, and ultimately derive an expected density (Thomas et al. 2010). Distance sampling

methods, including CTDS, assume unbiased and exact measurements. However, breaking the former may lead to larger estimate errors than breaking the latter assumption (Thompson 1994). REM employs ideal-gas models that require the camera trap's activation distance as an input (Rowcliffe et al. 2008) and it relies on accurate measurements (Palencia et al. 2021). This method hinges on accurate estimates of both the area of certain detection and the animals' travel distances (Rowcliffe et al. 2008), which can be derived from camera trap data. REST estimates density based on the amount of time animals spend within the camera's detection zone. This method relies on classifying detections within a specific area of the camera's view as "certain" (Nakashima et al. 2018). Biased distance measurements can lead to over or underestimation of density with REST. IS estimates the expected abundance within the camera's viewshed and extrapolates it (Moeller et al. 2018). Similar to Time To Event (TTE) from the same publication (Moeller et al. 2018), IS requires knowledge of the viewshed.

For all these methods, obtaining accurate distance measurements for analysis presents a challenge. To derive the camera activation distance in REM, researchers have employed various approaches. Some used a leashed cat as a proxy animal, walking it at different distances from the camera before deployment and annotating the activation (*e.g.* Manzo et al. 2012; Anile et al. 2014). Others have resorted to walking themselves in front of the camera (Carbajal-Borges et al. 2014). Or to assuming height of animals on the field according to their guessed age and combining this information with height ratios of bamboos photographed at different distances for obtaining an approximate distance (Pal et al. 2021). An accurate but labor-intensive approach involved analyzing captured images later and returning to the camera location with animal prints to measure perceived distances in the original image (Harris et al. 2020). Similarly, some have used reference video material with researchers walking at different distances to later compare it with the position of captured animals (*e.g.* Cappelle et al. 2019; Bessone et al. 2020).

A systematic approach was proposed by Caravaggi et al. (2016) who took reference photos of a "bamboo garden" - several reference bamboo sticks are placed on the nodes of a grid in the ground. They then used these reference images to manually generate a distance grid using image manipulation software (*e.g.* Bedson et al. 2021; Kolowski et al. 2021). Using this principle of reference grid, Wearn et al. (2022) expanded the concept with similar triangles of the "camera-pinhole" principle to generate a distance-to-pixel map. This method achieved high accuracy, but processing time remained high due to the use of 35 bamboo sticks per camera location. Lastly,

more recent advancements also include the development of semi-automatic and fully automated AI tools for image processing (Haucke et al. 2022; Johanns et al. 2022). While significantly improving processing speed (up to 21 times faster than traditional methods for the semi-automatic processes), these tools still exhibited error ranges between ±2m and ±7m for measurements up to 25 meters.

Thus a gap exists in the current methods for distance estimation of camera trap data. While many rely on large numbers of reference objects to manually draw accurate grids and some use basic perspective geometry or size ratio (*i.e.* the similar triangles of the camera pinhole principle), none employ the principles of size ratio and similar triangles to calibrate a grid that adjusts to desired distance intervals. This approach, utilizing perspective geometry, could significantly reduce the number of poles required in a reference image, accelerating the process. Furthermore, the Python Imaging Library (PIL) for image manipulation offers functionalities for pixel manipulation and text insertion. By combining these tools with Python scripting, researchers could readily generate perspective distance grids based on the pixel locations of reference objects within the image.

This work presents a novel method for estimating the position of lines on an image corresponding to increasing distances from the camera. The method uses the principles of similar triangles and geometric perspective to achieve accurate results while requiring only two reference objects, reducing man-hour resources required thus far. I propose a user-friendly software application developed in Python using the PIL library for image processing, ensuring compatibility across different operating systems. To validate the method's accuracy, I conducted a series of tests on images captured from varying distances and positions within the camera's field of view. The estimated distances obtained using my method were then compared to the real distances acquired with a measuring tape. Lastly, the estimate errors are analyzed.

Figure 1: Example of a result image with calibrated grid formed by horizontal lines for depth distance and the orthogonals for lateral distance estimation.

3 Methods

I developed a method to increase precision and reduce processing time for distance estimation of animals photographed with camera traps. The method consists of photographing two identical reference objects of known size at different distances in a camera setting as usual for monitoring animal presence. The reference photograph can then be run through an algorithm to draw grid consisting of a series of line segments indicating distances on the photographed landscape. The algorithm draws from algebra and linear and perspective geometry. It uses the furthest reference object to estimate the effective focal length of the camera using similar triangles (object-camera lens-camera sensor). The algorithm also uses the position of the two real reference objects on the image to find the vanishing point and create a perspective triangle that sets the boundaries

in which identical parallel objects would appear. The focal length is then used to estimate the hypothetical image size of a third object identical to the reference objects at a given distance. With the hypothetical image size and the perspective triangle, the image position of this hypothetical third object is estimated. The object is then prolonged to the edges of the image. The process is repeated for a fourth, fifth, and n-th object resulting in a sequence of horizontal lines representing equally increasing distances from the position of the camera. Combining the horizontal lines with the perspective triangle, a set of lateral-distance lines are represented on image to increase distance estimate precision.

3.1 Perspective Triangle

Using two parallel reference objects, that are also parallel to the camera's sensor, I wrote the algorithm to first create a reference frame consistent of a perspective triangle and a virtual coordinate system (VCS; Fig. 2). This reference frame is on the xy-plane formed by the cameras sensor and ultimately the photograph. Let the image of pole 1 (*i.e.* first reference object) start at a certain point A_1 and finish at a point further right B_1 and pole 2 (*i.e.* second reference object) start at A_2 and end at B_2. Then, the perspective triangle is limited by A_1, B_1 and the vanishing point (V_s). V_s is defined as the location, where the two straight lines, connecting A_1 with A_2 and B_1 with B_2 respectively, meet. In other terms, the perspective triangle is limited to the left and right by the linear equations A and B

$$y = m_A x + c_A \tag{A}$$

$$y = m_B x + c_B \tag{B}$$

respectively. The slope (m) and intercept (c) were determined for each equation by

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$c = y_1 - m x_1$$

where x_1, y_1, x_2 and y_2 represent the image-coordinates of pole 1 and 2 for side A or B. V_s is the point at which equations A and B meet. Downwards, the triangle is limited by the segment representing pole 1 on image. The perspective triangle determines the reference frame for later estimating the on-image length and position of a number N of new imaginary lines. Imaginary on-ground poles parallel to pole 1 have an image representation parallel to A_1B_1 and the start and end of the segments lies on lines A_1V_s and B_1V_s respectively.

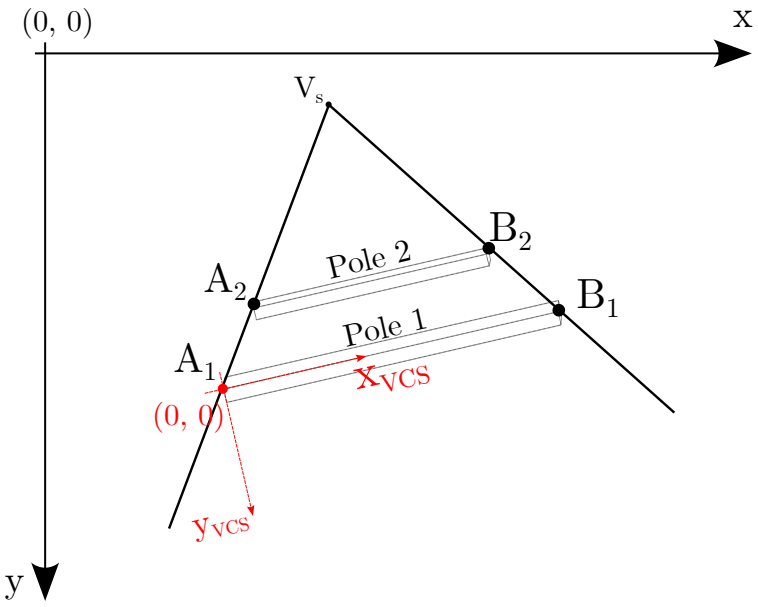

Figure 2: Perspective triangle and the image of poles 1 and two on the photographs xy-plane. The perspective triangle is built according to user-selected locations for the start and end of pole 1 (A_1, B_1; object closer to camera) and pole 2 (A_2, B_2; object further from camera). The algorithm uses these points to find the vanishing point (V_s) beyond which no distance can be estimated. The virtual coordinate system (red lines) is set with the origin at A_1 and the x-axis on pole 1.

3.2 Reference Grid for Distances on Image

3.2.1 Horizontal Lines

To find the size of the image of each new pole given a certain on-ground separation to the previous pole, I used similar triangles, Pythagora's theorem and a given separation distance d_s. Lets consider an above view of the camera setting (*i.e.* xz-plane, Fig. 3): there is the camera sensor, camera lens, one reference pole n and one new (imaginary) pole $n+1$. Both poles are parallel to each other and to the camera sensor. The new pole starts at a_{n+1} and ends at b_{n+1}, has the same length (w) as n and is represented on the camera's sensor by the segment $A_{n+1}B_{n+1}$ with length $l_{n+1} < l_n$. If we consider that the triangle formed between the new pole $a_{n+1}b_{n+1}$ and the camera lens, and

the one formed between the camera lens and the image of the new pole $B_{n+1}A_{n+1}lens$ are similar triangles (pink line on Fig. 3). We can use the real width of the objects w, the focal distance of the camera (*i.e.* distance between camera sensor and lens) f and the distance between the camera and the new pole d_{n+1} to get the size of the image of the new object l_{n+1} by

$$l_{n+1} = \frac{wf}{d_{n+1}}$$

Here we still need f, for which we can use the set of similar triangles formed by the camera lens, the original pole ($a_n b_n$) and its image ($A_n B_n$) of length l_n (blue lines on Fig. 3). After clearing for f

$$f = \frac{d_n l_n}{w}$$

we can replace in (3.2.1) and get

$$l_{n+1} = \frac{d_n l_n}{d_{n+1}}. \tag{1}$$

Given that the camera's lens and sensor are not at ground level, the similar triangles from the above view used to estimate on-camera object length are different to the actual similar triangles formed between the camera sensor, camera lens and the reference object (Fig. 4). Not accounting for this difference would result in a difference between the estimated length that considers on-ground triangles and the real length that is determined by the air-line triangles. To avoid this, I worked with the air-line distance between the camera and the current object (d_{air_n}, blue line on Fig. 4) and between the camera and the following object ($d_{air_{n+1}}$, pink line on Fig. 4) instead of the corresponding d_n and d_{n+1}. Using the separation d_s between each object (n) and the next one ($n+1$), the distance between the camera and the current object d_n (with which $d_n + d_s = d_{n+1}$) and the height from ground to the camera lens h_c. I estimated this distances with Pythagoras considering the rectangular triangle formed between the camera, the ground immediately below it and the corresponding object, using the equations

$$d_{air_n} = \sqrt{d_n^2 + h_c^2}, \tag{2}$$

and

$$d_{air_{n+1}} = \sqrt{(d_n + d_s)^2 + h_c^2}, \tag{3}$$

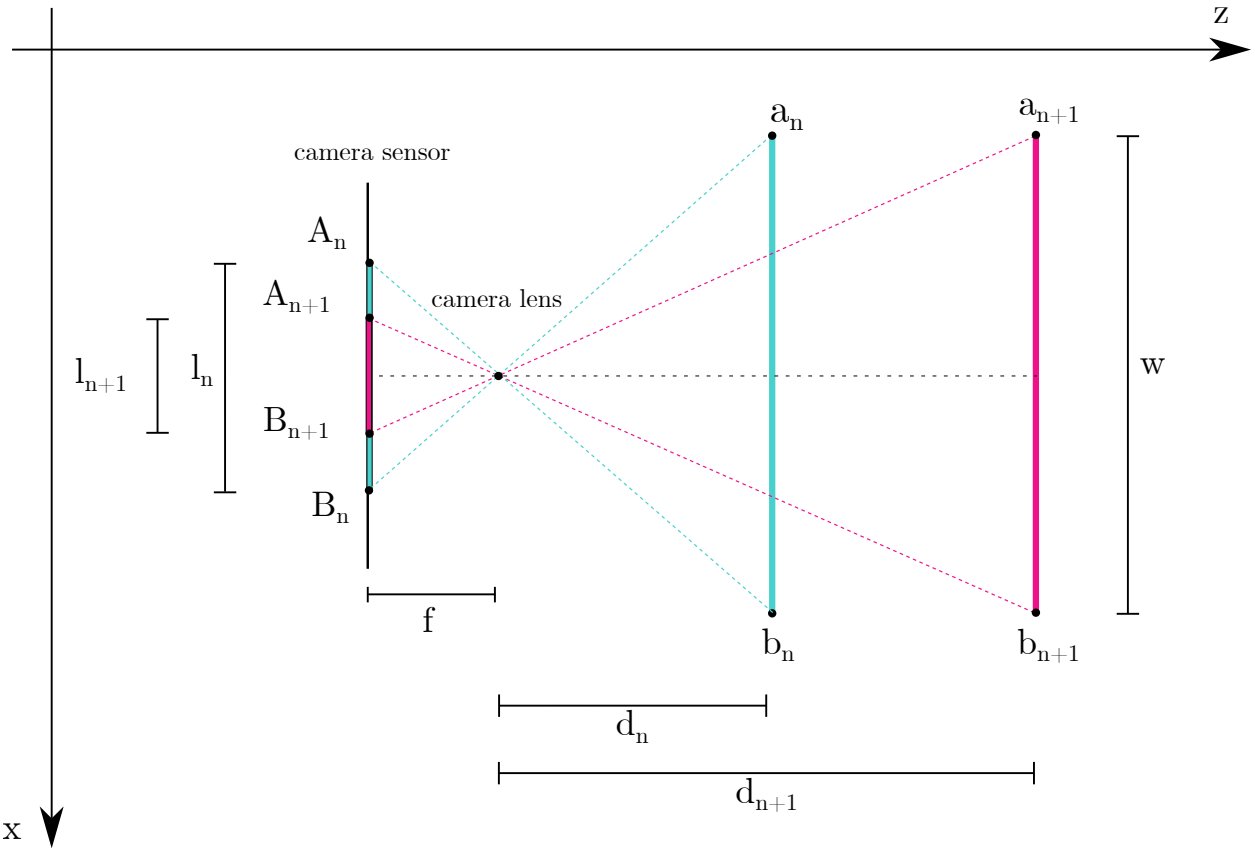

Figure 3: Top view of similar triangles used to estimate the size of an imaginary object. Using known parameters of a given reference object (blue) relative to the camera, the first set of similar triangles ($a_n b_n lens$ to $B_n A_n lens$) is used to estimate the image size (l_{n+1}) of a second object (pink) at a given distance from the camera (d_{n+1}). The assumption of parallelism between camera and reference object is set here.

To obtain the position on the image of each new hypothetical object and thus the necessary separations on image between lines of grid to reference real distances, I used the perspective triangle described above. Under the assumption that each new object (from here on poles) is parallel to pole 1, there is one and only one position inside the perspective triangle, where the length between segments A_1V_s and B_1V_s assumes a given value. In this case the given value is the length of the new pole (l_{n+1}, see Fig. 5). To obtain said position, I assumed that pole 1 and each new pole, were parallel to the x-axis (for cases in which this assumption is not directly met, see Virtual Coordinate System). If the segment representing a new pole $(n+1)$ has the limiting coordinates $x_{A_{n+1}}, y_{A_{n+1}}$ to the left and $x_{B_{n+1}}, y_{B_{n+1}}$ to the right, then the assumption of the image of the poles beeing

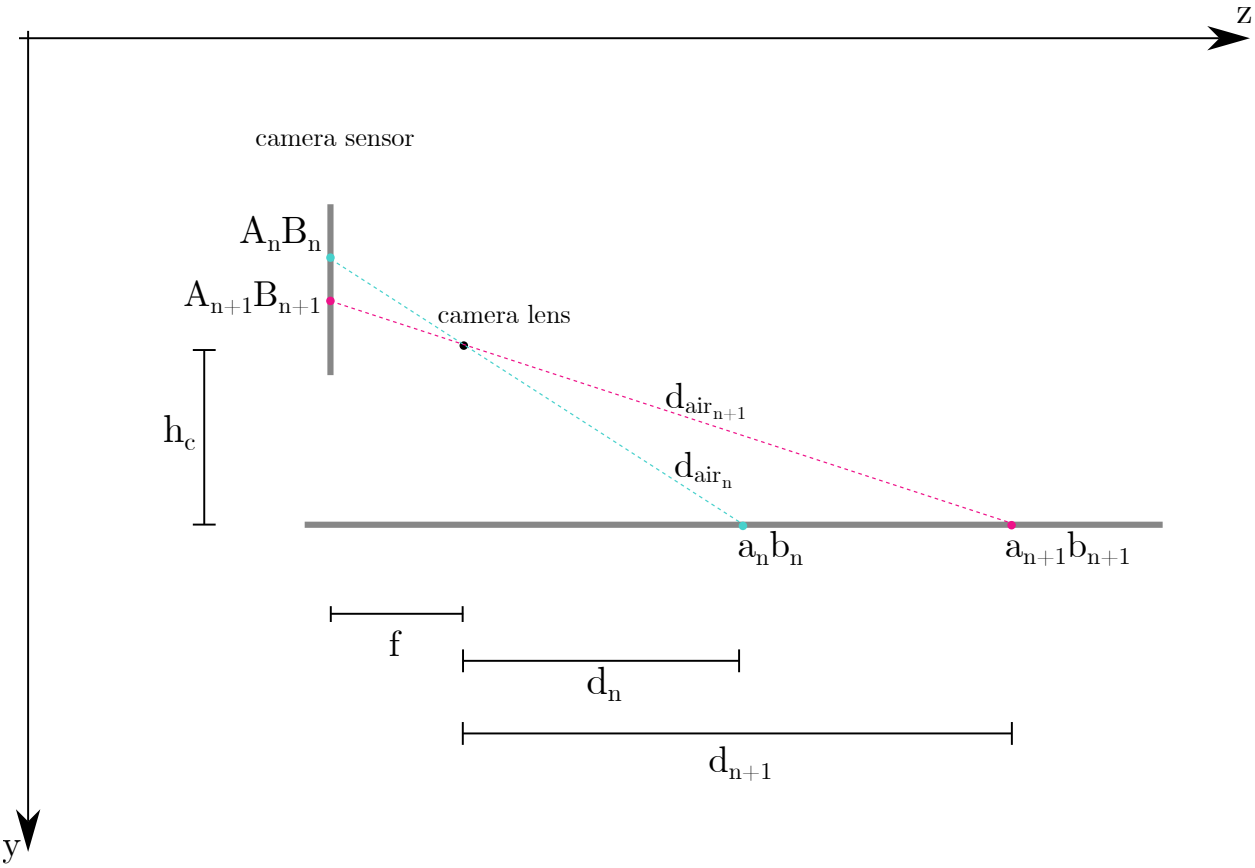

Figure 4: View from the side of similar triangles. This figure shows the sets of similar triangles for the object of known estimates (dashed blue line) and the object to be drawn on the image (dashed pink line). The points for the known object (A_nB_n) and the object to be drawn ($A_{n+1}B_{n+1}$) represent straight lines on the camera sensor or image file. The points for the known object (a_nb_n) and the object to be drawn ($a_{n+1}B_{n+1}$) represent straight lines on the ground. The real lengths of the air lines (d_{air_n}, $d_{air_{n+1}}$) are greater than if they were measured on the ground (d_n, d_{n+1}), which is reflected in the equation (3).

parallel means that $y_{A_{n+1}} = y_{B_{n+1}}$. Using (A) and (B) to find the value of these coordinates along the segments A and B, we get

$$m_A x_{A_{n+1}} + c_A = m_B x_{B_{n+1}} + c_B. \tag{4}$$

This, considering that the difference between $x_{B_{n+1}}$ and $x_{A_{n+1}}$ is the length obtained from (1), means that we can get the on-image start coordinate x of the new pole by

$$x_{An+1} = \frac{c_B - c_A + l_{n+1} m_B}{m_A - m_B}. \tag{5}$$

The remaining coordinates can then be obtained by replacing x_{An+1} in Equation (A) and the

resulting y in Equation (B). After using Equation (5) to obtain the position of each new imaginary line using its length, the on-image result is a series of short line segments marking the distances $d_n, d_{n+1}, ..., d_N$ inside the perspective triangle. To cover the whole picture area, I extended the segments to the ends of the image.

3.2.2 On-Image Representation of Lines Orthogonal to Poles 1 and 2

For objects that are not photographed on the center of the camera's field of vision, the distance estimated using horizontal lines looses precision with increasing separation from the center. To allow for a precise user estimation, I added line segments to the image that build the perspective representation of lines orthogonal to the reference objects on the xz-plane, building a perspective grid. On image, this meant new line segments were drawn that start at different points on the anchor line and all meet at the vanishing point (V_s).

To achieve this perspective grid built by the reference objects, the new horizontal lines and their xz-orthogonals, I assumed that a diagonal line connecting two corners of a rectangle extends to connect the corners of identical rectangles in its path. For the left hand side of the perspective triangle, the crossing point between the horizontal line $n + 2$ and the extended diagonal line connecting B_nA_{n+1} can be assumed as C_{n+2} with a corresponding point on the xz-plane c_{n+2}. Equally, the image line segment C (defined by the function $f(x)_z$; Fig. 5) has a corresponding line from the above perspective c, which runs orthogonal to a. For the right hand side, the diagonal changes to A_nB_{n+1} and produces the image line segment D and d from above. This step is repeated up to $n/2$, retrieving a total of n orthogonal lines split equally on each side of the perspective triangle.

3.3 Virtual Coordinate System

The algorithm thus far addresses the perspective representation of a grid for a case in which the images of poles 1 and 2 are parallel to the photographs x-axis. This case is limiting in use, because it assumes that the landscape has the same inclination around the z-axis than the camera. To allow for different inclinations of landscape and camera, I added a series of geometric transformations to the algorithm. The transformations changed any coordinate on the original image (x, y) to a corresponding coordinate (x_{VCS}, y_{VCS}) on a virtual coordinate system (VCS; Fig. 2). The VCS had its x-axis on the anchor line (A_1B_1) and its origin at A_1. The transformations started with a

translation of the coordinates from their values of the original coordinate system (OCS; x_{OCS} and y_{OCS}) to the VCS using a vector v (x_v, y_v) that starts at the origin of the OCS and ends at the origin of VCS (x_{A_1}, y_{A_1}).

$$x_T = x_{OCS} - x_v \tag{6}$$

$$y_T = y_{OCS} - y_v \tag{7}$$

To complete the transformation, this is followed by a rotation using the angle of rotation α between the original x-axis and the VCSs x-axis (A_1B_1)

$$x_{VCS} = x_T \cos(\alpha) - y_T \sin(\alpha) \tag{8}$$

$$y_{VCS} = x_T \sin(\alpha) + y_T \cos(\alpha) \tag{9}$$

Using the length of the image of pole 1 (l_1) as the adjacent side and the height formed by the y-difference between the coordinate systems $(y_{B_1} - y_{A_1})$ as the opposite side, we get the trigonometrical equation

$$\sin(\alpha) = \frac{y_{B_1} - y_{A_1}}{l_1}.$$

Thus the angle α can be found by

$$\alpha = -\arcsin(\frac{y_{B_1} - y_{A_1}}{l_1}).$$

By transforming the coordinates of pole 1 and 2, the transformed perspective triangle $(A_{1_{VCS}} B_{1_{VCS}} V_{s_{VCS}})$ formed a basis for finding the position of horizontal lines, without assuming equal z-axis inclination between camera and landscape. After finding the positions of each new line in the VCS, the coordinates only had to be transformed back to the real coordinate system of the image. Back-transformation involved first rotating a coordinate by the angle $2\pi\alpha$

with (8) and (9) and translating the result using the inverse vector $-v$ $(-x_v, -y_v)$ with (6) and (7).

3.4 Implementation

I implemented the perspective distance estimation method (Fig. 6) described above following a stepwise approach detailed in Supp. Mat. B. The approach also involved superimposing the grid on a user-selected series of images. The implementation was done with Python 3 (Van Rossum and Drake 2009) using the Python Imaging Library (PIL; Clark (2015)) for image processing tasks and Tkinter (Lundh 1999) for user interface development.

To improve the user experience and enable distance estimation on the remaining images of the series, I implemented two additional functionalities. First, when importing images, I resize the image to fill the available screen space. This allows for precise user interaction when defining the position of the reference object and fine-tuning the reference grid, while still maintaining a full view of the image. Secondly, once the grid was created, I resized the grid back to the original image dimensions. This allows the grid to be superimposed on all remaining images in the same series, maintaining the established positional references. Importantly, all images were saved in the PNG format to preserve image quality.

3.5 Model Assumptions

Accurate calculations in this system rely on several assumptions about the camera setup and environment. One crucial assumption is a coordinate system in which the x-axis runs horizontally and increases rightwards, the y-axis vertically and increases downwards, the z-axis increases in towards the photographed object. The violation of some of this assumptions could cause the calculations to be completely off. Where necessary, I implemented user input control to avoid negative values. (2) Additionally, the program assumes poles are parallel to the xy-plane, violation of this assumption will cause the grid not to fit the landscape on image and distance estimates to be off, but will go unnoticed to the program. (3) Poles 1 and 2 are assumed parallel to each other in all directions. Violation of this assumption makes calculations invalid, user input validation restricts larger differences in slope of both poles. (4) The reference grid's separation from the ground is assumed to match the distance of the pole tips, and (5) a right angle is assumed between

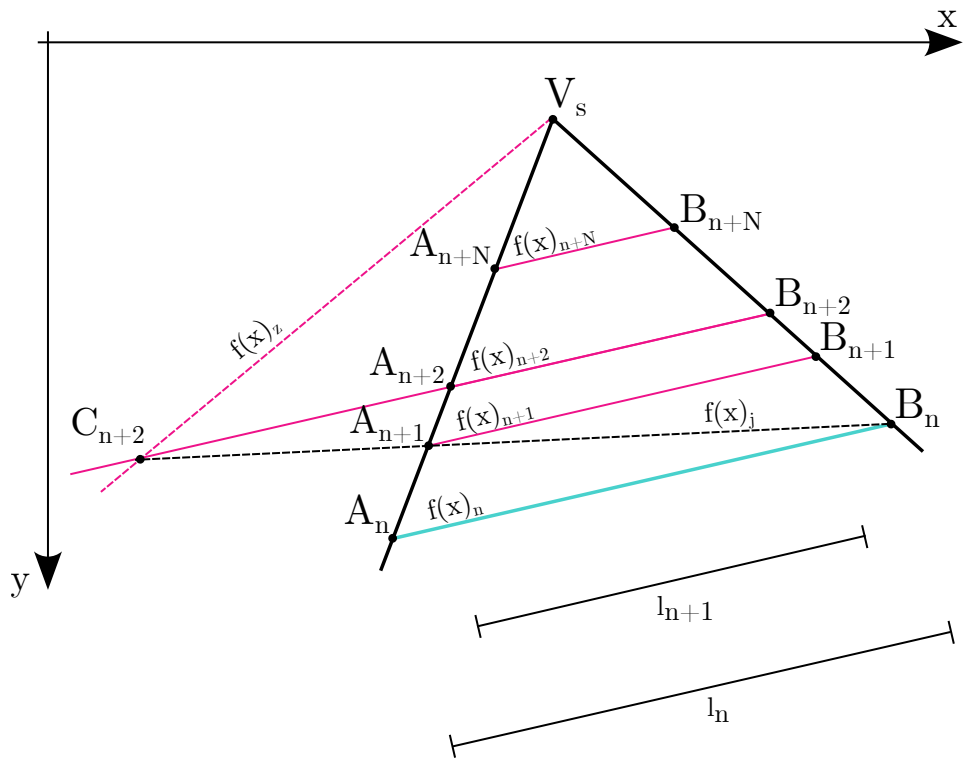

Figure 5: Perspective triangle with three depth-distance lines and visualization of lateral distance estimation. Using the perspective triangle and a reference object (full blue line; A_nB_n, $f(x)_n$), a sequence of parallel lines (full pink line; $f(x)_{n+1}, f(x)_{n+2}, ..., f(x)_{n+N}$) representing a given depth is found. In addition, the figure shows how a line marking the lateral distance (dashed pink line; $f(x)_z$) is equal to the length of the reference object. This is done by extending a diagonal (dashed black line) and the subsequent depth line, which gives the position of C_{n+2}.

the camera, object, and ground. This necessitates measuring the camera height precisely at a right angle to the ground. (6) Finally, the system assumes a static camera during its exposition time. To avoid problems caused by violating this assumption, calibration photos can be taken both before and after setup. (7) The assumption of a perfectly planar landscape requires empirical assessment, as deviations can significantly impact precision.

3.6 Case Study Estimation

The test photos were taken on a field with a low slope (2° inclination) and low vegetation. A Seissiger Special-Cam S308 camera trap was set to take 10 photographs of size "8 MP" (3264 × 2448) on a series after activated by the sensor set at "high" sensibility. The trigger intervall was set to 0 seconds. The camera trap was placed 0.43 m above the ground to photograph a reference object, which was later used for estimating the parameters of the digital distance grid. The reference object was 1.3 m wide and 1.0 m long, fitted with metal squares to ensure a right angle and placed

2.0 m away from the camera. To ensure that the reference object was parallel to the camera and in the middle of its FOV, the separation was measured at two points to the left and right of the camera. Within the same setting, a series of photographs were taken by placing activating the movement sensor and placing a chair representing a detected animal at several position. The chair was placed at distances of 2, 5 and 10 meters and at three different positions in the camera's FOV (center, side and edge of picture). For each of the nine positions, seven attempts of a photograph were made and the distance between the camera and the front left leg of the chair was measured with a measuring tape.

To estimate the distance between the camera and the detected "animal" using the above described digital perspective method, the 63 photographs were filtered to keep only those in which the object was on the ground and thus the measured distance corresponded to the photographed position. The photograph with the reference object was imported to the program and the distances of the setting were written as input. The position of the two reference poles was first pointed by clicking on their outer-most edges and corrected by visual comparison of the resulting grid with the horizon of the landscape to get the most parallel result possible. The corrected positions always remained within the frame marked by the poles. Given that the reference object defined the position of two lines at 2,0 m and 3,0 m, I set the separation to be represented between each new drawn line at 0,5 m. The total of lines was set at 19 to mark distances up to 11,0m from the x-axis of the camera. This maximum distance was set after visual evaluation of the drawn grid and the position of the chair through all images. The resulting reference frame was superimposed on the rest of the photo series.

To avoid obtaining biased distance perspective estimates because of the author's knowledge of the real measured distance, a student with no knowledge of the measurements performed perspective distance estimates. The distance estimation started by selecting a point of the selected chair leg at a height guessed to be equal to the height of the reference object above the ground. The position of this point between the horizontal lines was used to estimate a distance with a decimeter precision. This was considered the estimated z-distance. In a second step, the lateral distance to the center of the reference grid was guessed (x-distance). Finally, the resulting triangle formed by the z- and -distance was used to obtain the final perspective distance estimate between the chairs leg and the camera.

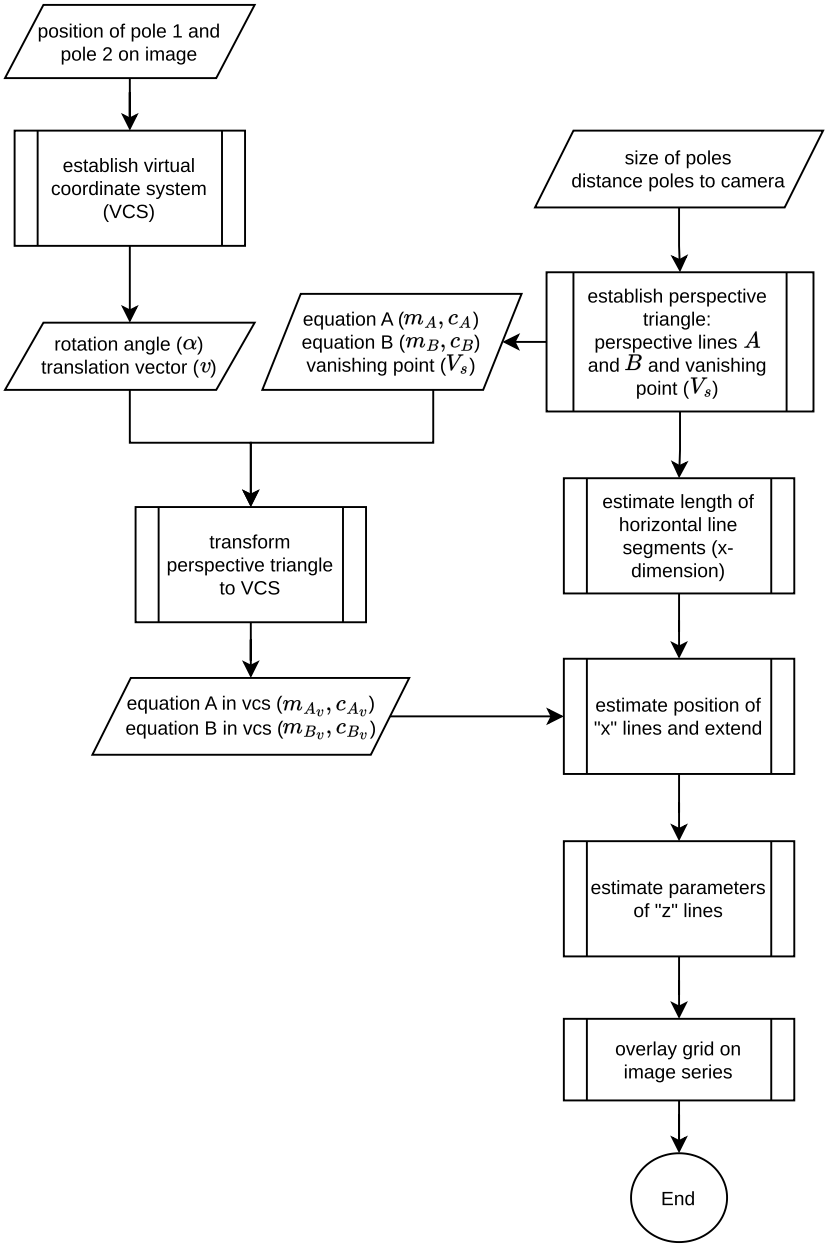

Figure 6: Flowchart of the distance grid calibration algorithm. Arrows indicate flow between steps, rectangles indicate a sub-process and parallelograms indicate user input. See Supp. Mat. Afor details of each sub-process.

3.7 Statistical analysis

To compare the distance measurements (M) and the perspective estimates (PE), I used Altman-Bland analysis (Altman and Bland 1983; Bland and Altman 1986). The method is used in medical studies to compare if one measurement method is an appropriate replacement of another or to compare it against a "golden standard". I compared the estimate error (*i.e.* difference of PE - M) with the distance measurement. Given the distribution of the estimate error strongly deviated from normality (Shapiro-Wilk normality test $p = 7.1\text{e-}05$) and the variation of residuals increased with increasing distance, I log transformed the measurement and the perspective estimate pre-analysis. I chose the log-transformation instead of the non-parametric approach suggested by Bland and Altman (1999), because the non-parametric alternative is less reliable for small sample sizes (here $n = 43$; Bland and Altman (1999)). The difference $(\ln(PE) - \ln(M))$ values was used to estimate the limits of agreement (mean ± 1.96 SD), which indicate that 95% of the errors should fall within this range. I then back-transformed the limits of agreement to get a geometric ratio of error to distance mean. Even though correlation coefficients are not an appropriate tool to detect systematic bias when comparing methods of measurement, I estimated Spearman's correlation coefficient ρ to compare the random error in this distance estimate method with other methods.

The relationship between increasing distance between object and camera and changes in position of the object respective to the cameras FOV and the estimate error (PE - M) were investigated by fitting three linear regression models with estimate error as the dependent variable. The first model only included distance (2 m, 5 m and 10 m) as a categorical variable, the second added the position in the cameras FOV and the third also included the interactions between measured distance and position in the FOV. The models were fitted using the method of least squares. To assess if the independent variables contributed significantly to explaining the variation of the dependent variable, sequential comparisons of the first, second and third model were performed (Table 1) using the sum of squares Type II (Fox and Weisberg 2019) because of the unbalanced collected data. To further evaluate the goodness of fit, the Akaike Information Criterion (AIC; Akaike (1973)) for each model was estimated. A factorial analysis of variance (ANOVA) was used to analyze the independent variables and their interaction on the estimate error. Finally, a Post-hoc analysis of the predictors was carried out using Bonferroni's method adjusted pairwise comparison of marginal means for 36 tests ($\alpha = 0.01$). For reporting results of the statistical analysis, I avoided reporting on "significance" where possible and opted for interpreting the p-values in their context

Table 1: Summary of the sequential F-tests of model comparisons used to assess the additive (Model 2 1) and interaction contribution (Model 3 2) of explanatory variables. The table reports the sum of squares, degrees of freedom (Df), F-value, and P-value for the each term and residuals. The analysis employed a linear regression framework with least squares estimation and unbalanced data considerations using Type II sum of squares.

Factor	Sum of sq.	Df	F Value	Pr(<F)
Model 2 ~ 1				
FOV Position	4.75	2	8.6	8e-04
Distance	4.79	2	8.66	7e-04
Residuals	11.05	40		
Model 3 ~ 2				
Distance	4.79	2	14.43	2.0e-05
FOV Position	4.75	2	14.32	2.0e-05
Distance:FOV Position	4.85	4	7.31	1.8e-04
Residuals	6.30	38		

(Amrhein et al. 2019; Di Leo and Sardanelli 2020). All statistical analyses were performed using the statistical software R (version 4.3.3, R Core Team (2024)).

4 Results

4.1 Software validation

The user interface of the program (Fig. 7.a) consisted of four command sections divided along the four input steps: reference file selection, reference input definition, grid settings, series output selection. The user must first select a file of the camera trap photo with the reference object composed of pole 1 (closer to the camera) and pole 2 (further away). When an image has been selected, it is displayed for visual reference. In the second step, the reference input requires indicating the width of the poles and separation of each one to the camera's position, as well as the height of the camera (Fig. 7.b). Clicking on each of the buttons "Point pole 1" and "Point pole 2" activates a mouse input that expects a sequence of two clicks on the image, one at each end of the respective pole. On the third step, the grid settings are selected by indicating the number of horizontal lines to be drawn and the distance to separate one line from the next one. The grid can be previewed (Fig. 7.c) to assure it fits references in the landscape, which can be a third reference object at a known distance or the horizon, if the grid does not fit, readjustment of parameters or reclicking the position of the poles is reccomended. Once the grid has been calibrated, the last step is to select the path to the rest of the images taken from the same camera position. The program will create a copy of each photo and superimpose the drawn grid on top each one.

4.2 Distances estimated from calibrated grid

The reference grid resulting from the perspective algorithm drew 18 reference lines that indicated depth distances between two and 11 m away from the camera's position. Additionally, 18 reference z-lines indicated lateral distances up to 4.15 m to the left and right of the center of the FOV, the z-lines were separated by 0.5 m. The grid was superimposed on 43 images (*e.g.* 1) and the distance to the photographed chair was estimated (*e.g.* Fig. 7.d).

A total of 43 points were analysed for the estimate error between perspective-estimated and measured distance with a mean error of -0.17 m. Comparison of linear models with increasing number of independent variables showed high F-statistics for all variables (Table 1). The results indicated that both FOV position (F = 8.60, p = 8e^-04) and categorical distance (F = 8.66, p = 7e^-04) contributed to explaining the variation in the estimate error. In addition, the interaction factor

(FOV position x categorical distance, $F = 7.31$, $p = 1.8\text{e-}04$) also a large contribution. This suggests that the effect of FOV position on the estimate error depends on the specific category of distance, and vice versa. The third model emerged as the best fit based on two criteria: the lowest AIC value ($\Delta_{m_3-m2} = -55.1$; $\Delta_{m_3-m1} = -75.2$) and the highest adjusted R^2 (0.89), meaning that it explained a high percentage of the variation in the data. The third model was thus selected for further analysis. The selected model predicted that the magnitude of the error would vary as a function of the interaction between position (center, side or edge) and distance (2m, 5m, or 10m). As shown in Fig. 8, the center position at 10m distance had the largest predicted negative error (-1.3m), suggesting a tendency to underestimate distance. In contrast, the error was minimal for the center position at both 2m (-0.04m) and 5m (-0.10m), indicating good agreement between estimated and measured distances.

The details of these relationships were illustrated by the pairwise comparisons of the distance error between the 9 factor groups (three FOV positions x three distance categories). The Bonferroni corrected comparison revealed three distinct groups (Fig. 8). The first group included the three FOV-factors at 2m and 5m, as the measurement error of all six factors had compatible intervals (Di Leo and Sardanelli 2020). The measurement errors of this group were close to zero with mean values between -0.03 (2 m x center) and 0.21 (2 m x edge). The second group included center and side of the FOV at 10m, factors which had a mean error of -1.37 m and of -1.03 m suggesting an underestimation of the true distance. The third group consisted of measurements at the FOV edge in all distance categories and showed a slight pattern of overestimation.

Bland–Altman plots for the distance measurements and estimates (Fig. 9) suggested an overall agreement between the measured and perspective-estimated distances. The geometric mean ratio of values by the measurement and estimation method was 0.99 with 95% limits of agreement at 0.85 to 1.16. Thus the perspective-estimation method differed from tape measurement on a ratio of 0.99 ($\pm$0.85 and 1.16). Spearmans correlation test between the measured and the estimated distance resulted in $\rho = 0.88$ ($p = 7^{-15}$).

Figure 7: Start window and grid calibration steps for perspective distance estimation. (a) The start menu with empty input ready file selection of the reference image. (b) User interface after entering the parameters of the reference objects and the position of pole 1 with a two-click system, the position of pole 2 must also be entered. (c) Visual preview of the grid after entering the drawing parameters: number of lines and line spacing. After the visual preview, step b can be repeated to adjust the references in the landscape. (d) Superimposed grid on an image of the series with distance and metadata. The image also shows an example of estimating the distance between the right leg of the chair: for a frontal distance of 2.0 m and a lateral distance of 0.7 m, the total perspective estimated distance x is 2.1 m. The user interface was developed using the Tkinter library in Python.

5 Discussion

5.1 Method

I developed a distance estimation method for camera trap images that comprises a user-friendly software. The method relies on similar triangles and perspective geometry to approximate the representation of real distances on image. For the users the estimation of distance for camera-trapped animals relies on a three-step process:

1. **Camera Setup**: The camera is positioned with a reference object placed within the frame Supp. Mat. C. This object consists of two poles with known dimensions arranged at a right angle parallel to the camera. See Model Assumptions for a detailed description of the assumptions on this setup. Alternatively, four bamboos can be used as reference objects, with additional measurements to avoid breaking assumptions.
2. **Manual Input, Preview, and Superimpose**: This stage involves user input of necessary for calibration including click-location of the reference objects on image and a visual confirmation of the resulting grid.
3. **Distance Estimation from Grid**: Users locate the detected animal and select a specific pixel of interest in the image. The superimposed grid aids in estimating both the depth and lateral distances of the animal from the camera. Finally, the users calculate the direct distance from the depth and lateral distance using the three-squares theorem.

The software program for implementing this method is freely available on GitHub (GitHub) and is compatible with Windows, Linux, and Mac operating systems. It requires Python 3 or above to be executed and instructions for setup and execution can be found in the GitHub repository. The user interface is designed for ease of use and facilitates distance estimation for researchers with varying technical backgrounds.

5.2 Estimation Errors

The results indicate that distance estimates were accurate at close range but became less precise as the distance between the camera and the object increased. The average error was a small underestimation of 0.17 m, with errors ranging from a 15% underestimation to a 16% overestimation.

This decrease in precision with increasing distance is consistent with theoretical expectations and similar studies (Caravaggi et al. 2016; Howe et al. 2017; Johanns et al. 2022). At greater distances, a smaller area of the image represents the same actual space compared to closer objects. This inherent perspective limitation leads to increases in uncertainty for distance measurements proportional to distance.

In comparison to similar studies, my results show comparable performance. Caravaggi et al. (2016) achieved high accuracy of estimated distance with the measured distance as an explanatory variable having an R^2 value of 0.98 using a calibrated digital grid with reference poles. They reported a significant correlation, but given that they report no correlation coefficient the comparison is difficult. They also reported a decrease in explained variation by their model ($R^2 = 0.75$) for measurements greater than 6 m, indicating, similar to my results, a decrease in precision at greater distances. Their absolute mean error (0.22 m) was very close to mine, suggesting overall equivalence in the precision of both methods.

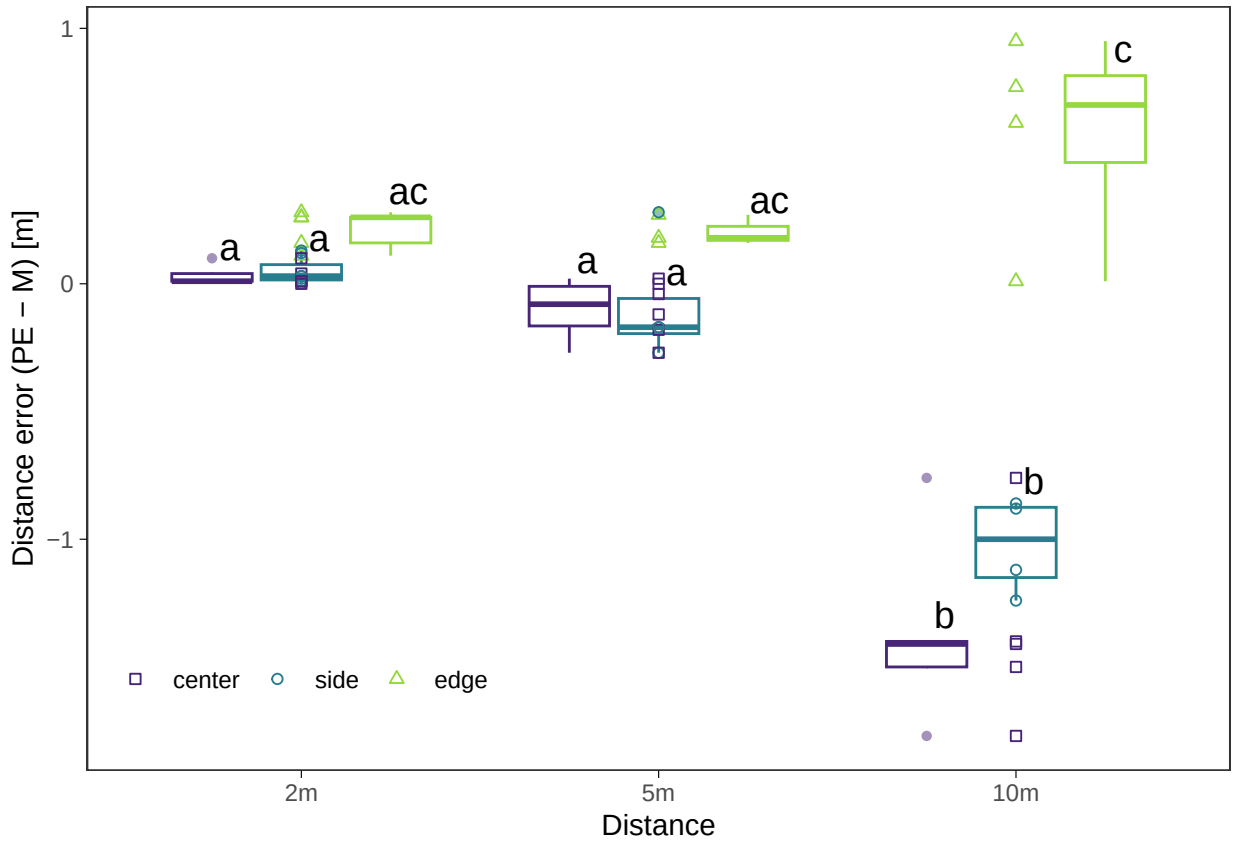

Figure 8: Effects of the measured distance and FOV position on the distance error from perspective estimation. Bars sharing the same letter do not differ significantly, Bonferroni adjusted pairwise test of marginal means ($p > 0.01$); means ±SD (boxes), range of minimum to maximum value (vertical lines) and outliers (solid circles).

In contrast, Haucke et al. (2022) report that their two-step AI approach achieved a mean relative error of 0.04 at 2 m and 0.01 at 9 m, indicating higher precision at larger distances. Their results show a higher precision than mine, although the mean error (± 2.5 m) varied considerably between camera locations and error values ranged widely (± 7.5 m). They explained some of the differences by errors in the automated animal image location, by which background pixels were identified as the detected animal, therefore stretching the estimate (Haucke et al. 2022). Furthermore, the fully automated AI method proposed by Johanns et al. (2022) had a higher absolute error (0.99 m) than mine (0.17 m), but they had measurements going up to 25 m and their relative error (0.113) was similar to mine. The combination of high precision and a larger range of error in the AI methods suggest, that the perspective-estimation method has a higher reliability. However, it should be considered that Haucke et al. (2022) and Johanns et al. (2022) had data sets larger by several orders of magnitude.

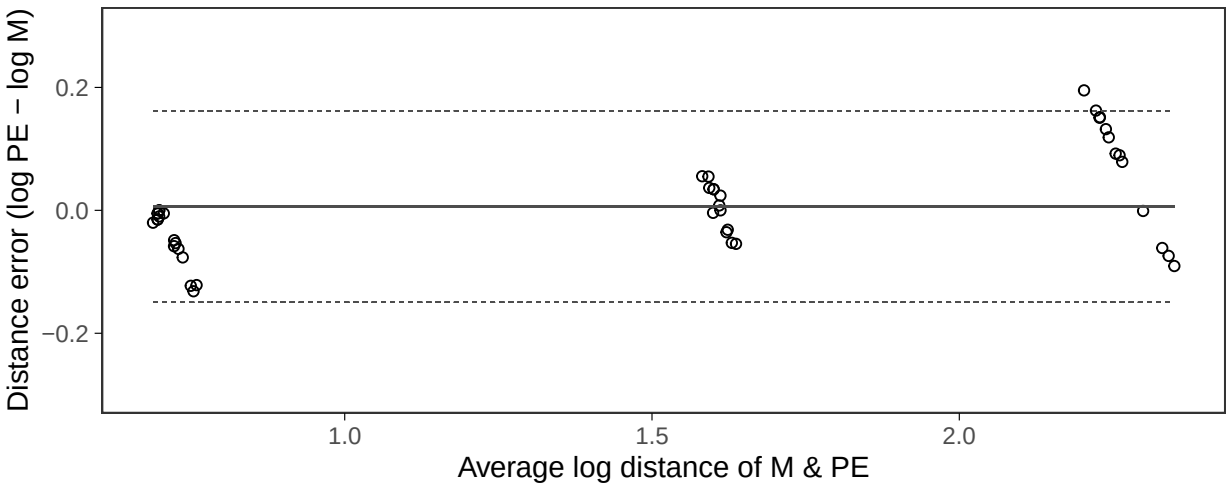

Figure 9: Bland-Altman plot to assess the agreement between distance measurements (M) and perspective estimates (PE). The estimate error (PE-M) is plotted against the log average of M and PE. The geometric mean ratio (solid line) between measured and estimated values was 0.99 and the 95% limits of agreement (dashed lines) ranged from 0.85 to 1.16.

My method is expected to have an error proportional to distance, with an anticipated range of ±15%. While the Bland-Altman plot method can define intervals of agreement, acceptable error limits ultimately depend on the specific research question (Altman and Bland 1983). Determining acceptable limits would involve comparing the impact of different measurement methods on population density estimates. In this respect, Caravaggi et al. (2016) investigated the relationship between measurement methods and density estimates for Irish hares. They found no significant difference between estimates derived from real, measured distances and those estimated using a 15-pole digital grid method. Again, my result cannot easily be compared to theirs due to their missing correlation or ratio parameters (Caravaggi et al. 2016). Additionally, the theoretical assumptions of models can provide insight into the importance of unbiased and exact measurements. For instance, methods like CTDS and REM assume exact measurements, while others like TTE, STE and IS are less reliant on this assumption (Thompson 1994; Rowcliffe et al. 2008; Nakashima et al. 2018; Palencia et al. 2021).

This study revealed an additional source of error related to the field of view (FOV). When the target object was positioned further from the image center, particularly at distances exceeding 10 meters, the error range increased. This can be explained by the interplay between grid tilt and object position within the FOV. Ideally, a horizontal grid line should pass through a specific point in the image. However, any tilt in the grid will cause this line to miss the target position, with the magnitude of error increasing as the line deviates further from the FOV center. The lack of a calibration reference for grid tilt contributed to larger errors. Furthermore, the error caused by the object's position within the FOV also grew with increasing distance from the center. This combined effect arises from the interplay of two error sources: an inaccurate grid inclination and the fact that pixels represent a larger area at greater distances. This phenomenon is supported by the observation of larger errors for objects positioned on the right edge of the image compared to the center. These findings highlight the importance of visual calibration by the user to ensure a correct grid alignment. A practical solution could involve incorporating a fifth reference point with known angle and separation (in addition to the four ends of poles 1 and 2) to aid in visually selecting the correct click positions on the reference poles.

Apart from the empirical and practical error sources, the method proposed above contains a conceptual limitation. The method assumes the focal distance to stay identical for the complete image, however the different angle between an object at the center and at the edge of the image logically

increases the distance between the pixel and the camera lense. If this difference is not negligible, the assumption could potentially introduce another source of error. Additionally, while several existing methods explore video data for distance estimation, my software's current limitation lies in its reliance on the Python Imaging Library (PIL) for raster processing, which prevents grid superimposition on video footage. However, this functionality could be a promising avenue for future software development.

In conclusion, the presented method allows researchers to easily estimate distance to animals in camera trap photographs. The error is comparable to usual methods and somewhat smaller than recently proposed AI-trained algorithms. The processing time should be faster than currently used methods, but slower than the AI-trained algorithms.

6 References

Akaike, H. 1973. Maximum likelihood identification of gaussian autoregressive moving average models. Biometrika 60:255–265. Oxford University Press.

Altman, D. G., and J. M. Bland. 1983. Measurement in medicine: The analysis of method comparison studies. Journal of the Royal Statistical Society Series D: The Statistician 32:307–317. Oxford University Press.

Amrhein, V., S. Greenland, and B. McShane. 2019. Scientists rise up against statistical significance. Nature 567:305–307. Nature Publishing Group UK London.

Anile, S., B. Ragni, E. Randi, F. Mattucci, and F. Rovero. 2014. Wildcat population density on the e tna volcano, i taly: A comparison of density estimation methods. Journal of Zoology 293:252–261. Wiley Online Library.

Bedson, C. P., L. Thomas, P. M. Wheeler, N. Reid, W. E. Harris, H. Lloyd, D. Mallon, and R. Preziosi. 2021. Estimating density of mountain hares using distance sampling: A comparison of daylight visual surveys, night-time thermal imaging and camera traps. Wildlife Biology 2021:wlb–00802. Wiley Online Library.

Bessone, M., H. S. Kühl, G. Hohmann, I. Herbinger, K. P. N'Goran, P. Asanzi, P. B. Da Costa, V. Dérozier, E. D. Fotsing, B. I. Beka, and others. 2020. Drawn out of the shadows: Surveying secretive forest species with camera trap distance sampling. Journal of Applied Ecology 57:963–974. Wiley Online Library.

Bland, J. M., and D. Altman. 1986. Statistical methods for assessing agreement between two methods of clinical measurement. The lancet 327:307–310. Elsevier.

Bland, J. M., and D. G. Altman. 1999. Measuring agreement in method comparison studies. Statistical Methods in Medical Research 8:135–160.

Brand, U., and M. Wissen. 2013. Crisis and continuity of capitalist society-nature relationships: The imperial mode of living and the limits to environmental governance. Review of International Political Economy 20:687–711. Taylor & Francis.

Cain, M. L., W. D. Bowman, and S. D. Hacker. 2011. Ecology. Oxford University Press, Incorporated.

Cappelle, N., M.-L. Després-Einspenner, E. J. Howe, C. Boesch, and H. S. Kühl. 2019. Validating camera trap distance sampling for chimpanzees. American Journal of Primatology 81:e22962. Wiley Online Library.

Caravaggi, A., M. Zaccaroni, F. Riga, S. C. Schai-Braun, J. T. Dick, W. I. Montgomery, and N. Reid. 2016. An invasive-native mammalian species replacement process captured by camera trap survey random encounter models. Remote Sensing in Ecology and Conservation 2:45–58. Wiley Online Library.

Carbajal-Borges, J. P., O. Godínez-Gómez, and E. Mendoza. 2014. Density, abundance and activity patterns of the endangered tapirus bairdii in one of its last strongholds in southern mexico. Tropical Conservation Science 7:100–114. SAGE Publications Sage CA: Los Angeles, CA.

Ceballos, G., P. R. Ehrlich, A. D. Barnosky, A. García, R. M. Pringle, and T. M. Palmer. 2015. Accelerated modern human–induced species losses: Entering the sixth mass extinction. Science Advances 1:e1400253.

Clark, A. 2015. Pillow (PIL fork) documentation. readthedocs.

Corlatti, L., S. Sivieri, B. Sudolska, S. Giacomelli, and L. Pedrotti. 2020. A field test of unconventional camera trap distance sampling to estimate abundance of marmot populations. Wildlife Biology 2020:1–11. Wiley Online Library.

Di Leo, G., and F. Sardanelli. 2020. Statistical significance: P value, 0.05 threshold, and applications to radiomics—reasons for a conservative approach. European radiology experimental 4:1–8. Springer.

Díaz, S., J. Settele, E. S. Brondízio, H. T. Ngo, J. Agard, A. Arneth, P. Balvanera, K. A. Brauman, S. H. Butchart, K. M. Chan, and others. 2019. Pervasive human-driven decline of life on earth points to the need for transformative change. Science 366:eaax3100. American Association for the Advancement of Science.

Foster, R. J., and B. J. Harmsen. 2012. A critique of density estimation from camera-trap data. The Journal of Wildlife Management 76:224–236. Wiley Online Library.

Fox, J., and S. Weisberg. 2019. An R companion to applied regression. Third. Sage, Thousand Oaks CA.

Gilbert, N. A., J. D. Clare, J. L. Stenglein, and B. Zuckerberg. 2021. Abundance estimation of unmarked animals based on camera-trap data. Conservation Biology 35:88–100. Wiley Online Library.

Griffiths, M., and C. P. Van Schaik. 1993. The impact of human traffic on the abundance and activity periods of sumatran rain forest wildlife. Conservation Biology 7:623–626. JSTOR.

Harris, G. M., M. J. Butler, D. R. Stewart, E. M. Rominger, and C. Q. Ruhl. 2020. Accurate

population estimation of caprinae using camera traps and distance sampling. Scientific Reports 10:17729. Nature Publishing Group UK London.

Haucke, T., H. S. Kühl, J. Hoyer, and V. Steinhage. 2022. Overcoming the distance estimation bottleneck in estimating animal abundance with camera traps. Ecological Informatics 68:101536. Elsevier.

Hay, T. 2024. Simulation-based evaluation of density estimators for unmarked animals.

Howe, E. J., S. T. Buckland, M.-L. Després-Einspenner, and H. S. Kühl. 2017. Distance sampling with camera traps. Methods in Ecology and Evolution 8:1558–1565. Wiley Online Library.

Johanns, P., T. Haucke, and V. Steinhage. 2022. Automated distance estimation for wildlife camera trapping. Ecological Informatics 70:101734. Elsevier.

Karanth, K. U., and J. D. Nichols. 1998. Estimation of tiger densities in india using photographic captures and recaptures. Ecology 79:2852–2862. Wiley Online Library.

Kissling, M. L., and E. O. Garton. 2006. Estimating detection probability and density from point-count surveys: A combination of distance and double-observer sampling. The Auk 123:735–752.

Kolowski, J. M., J. Oley, and W. J. McShea. 2021. High-density camera trap grid reveals lack of consistency in detection and capture rates across space and time. Ecosphere 12:e03350. Wiley Online Library.

Kühl, H. S., S. T. Buckland, M. Henrich, E. Howe, and M. Heurich. 2023. Estimating effective survey duration in camera trap distance sampling surveys. Ecology and Evolution 13:e10599. Wiley Online Library.

Lundh, F. 1999. An introduction to tkinter. URL: www. pythonware. com/library/tkinter/introduction/index. htm.

Manzo, E., P. Bartolommei, J. M. Rowcliffe, and R. Cozzolino. 2012. Estimation of population density of european pine marten in central italy using camera trapping. Acta Theriologica 57:165–172. Springer.

McRae, L., R. Freeman, J. Geldmann, G. B. Moss, L. Kjær-Hansen, and N. D. Burgess. 2022. A global indicator of utilized wildlife populations: Regional trends and the impact of management. One Earth 5:422–433. Elsevier.

Moeller, A. K., P. M. Lukacs, and J. S. Horne. 2018. Three novel methods to estimate abundance of unmarked animals using remote cameras. Ecosphere 9:e02331. Wiley Online Library.

Moore, J. W. 2016. Anthropocene or capitalocene?: Nature, history, and the crisis of capitalism. Pm Press.

Nakashima, Y., K. Fukasawa, and H. Samejima. 2018. Estimating animal density without individual recognition using information derivable exclusively from camera traps. Journal of Applied Ecology 55:735–744. Wiley Online Library.

Otis, D., K. Burnham, G. White, and D. Anderson. 1978. Statistical inference from capture data on closed animal populations, volume 62. Wildlife society.

Pal, R., T. Bhattacharya, Q. Qureshi, S. T. Buckland, and S. Sathyakumar. 2021. Using distance sampling with camera traps to estimate the density of group-living and solitary mountain ungulates. Oryx 55:668–676. Cambridge University Press.

Palencia, P., J. Fernández-López, J. Vicente, and P. Acevedo. 2021. Innovations in movement and behavioural ecology from camera traps: Day range as model parameter. Methods in Ecology and Evolution 12:1201–1212. Wiley Online Library.

R Core Team. 2024. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

Rowcliffe, J. M., J. Field, S. T. Turvey, and C. Carbone. 2008. Estimating animal density using camera traps without the need for individual recognition. Journal of Applied Ecology 1228–1236. JSTOR.

Thomas, L., S. T. Buckland, E. A. Rexstad, J. L. Laake, S. Strindberg, S. L. Hedley, J. R. Bishop, T. A. Marques, and K. P. Burnham. 2010. Distance software: Design and analysis of distance sampling surveys for estimating population size. Journal of Applied Ecology 47:5–14. Wiley Online Library.

Thompson, S. K. 1994. Biometrics 50:891–892. [Wiley, International Biometric Society].

Valente, A. M., P. Acevedo, A. M. Figueiredo, C. Fonseca, and R. T. Torres. 2020. Overabundant wild ungulate populations in europe: Management with consideration of socio-ecological consequences. Mammal Review 50:353–366. Wiley Online Library.

Van Rossum, G., and F. L. Drake. 2009. Python 3 reference manual. CreateSpace, Scotts Valley, CA.

Wearn, O. R., T. E. Bell, A. Bolitho, J. Durrant, J. K. Haysom, S. Nijhawan, J. Thorley, and J. M. Rowcliffe. 2022. Estimating animal density for a community of species using information obtained only from camera-traps. Methods in Ecology and Evolution 13:2248–2261. Wiley Online Library.

Wilson, K. R., and D. R. Anderson. 1985. Evaluation of two density estimators of small mammal population size. Journal of Mammalogy 66:13–21. American Society of Mammalogists, 810

East 10th Street, PO Box 1897, Lawrence.

7 Supporting Materials

7.1 Supporting Material A

Table 2: Summary of the Bonferroni adjusted pairwise test of marginal means between group means for distance and FOV-position groups. The table reports the groups mean estimate, standard error, F-value, and P-value for the each term and residuals.

Factor	Estimate	Std. error	F Value	Pr(<F)
2 m : center	0.032	0.092	0.346	7.3e-01
5 m	-0.130	0.125	-1.041	3.1e-01
10 m	-1.400	0.131	-10.706	2.0e-12
Side	0.017	0.121	0.137	8.9e-01
Edge	0.182	0.131	1.392	1.7e-01
5 m : side	-0.001	0.180	-0.004	1.0e+00
10 m : side	0.326	0.184	1.773	8.5e-02
5 m : edge	0.120	0.196	0.610	5.5e-01
10 m : edge	1.776	0.191	9.317	6.9e-11

7.2 Supporting Material B

7.2.1 B.1

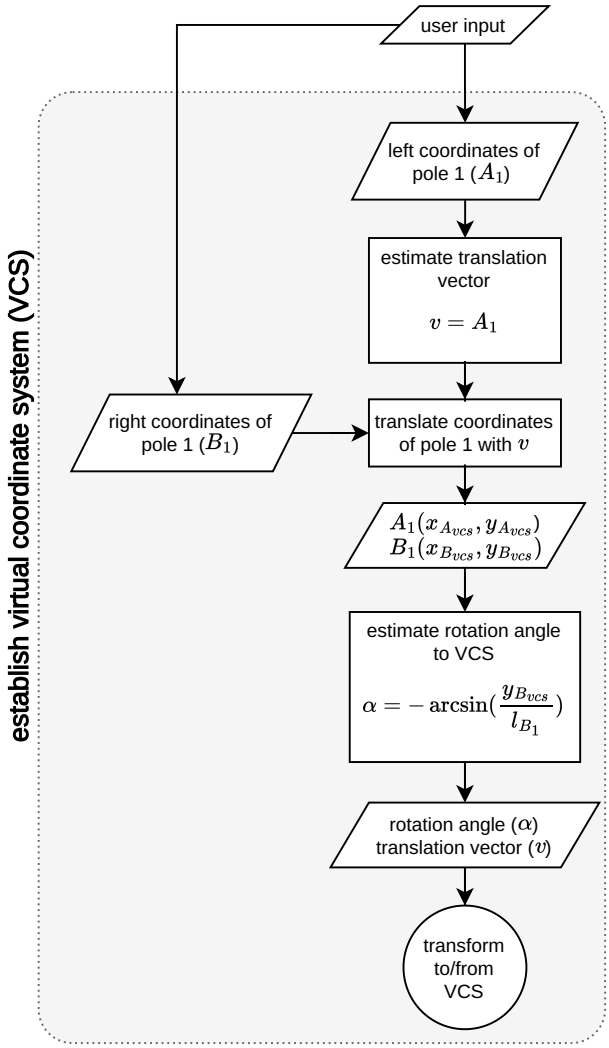

Figure 10: Flowchart for creation of the virtual coordinate system. This represents step 1 of the programm after user input.

7.2.2 B.2:

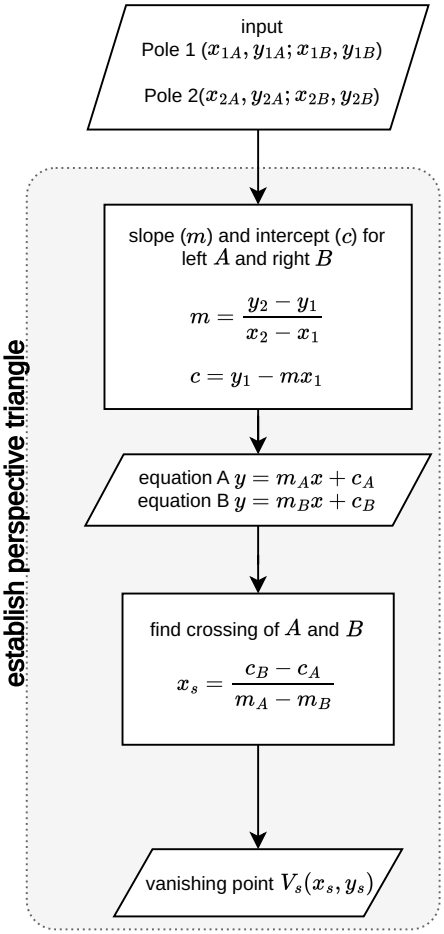

Figure 11: Flowchart for establishing the perspective triangle. This represents step 2 of the program is to establish the perspective triangle on image to .

7.2.3 B.3

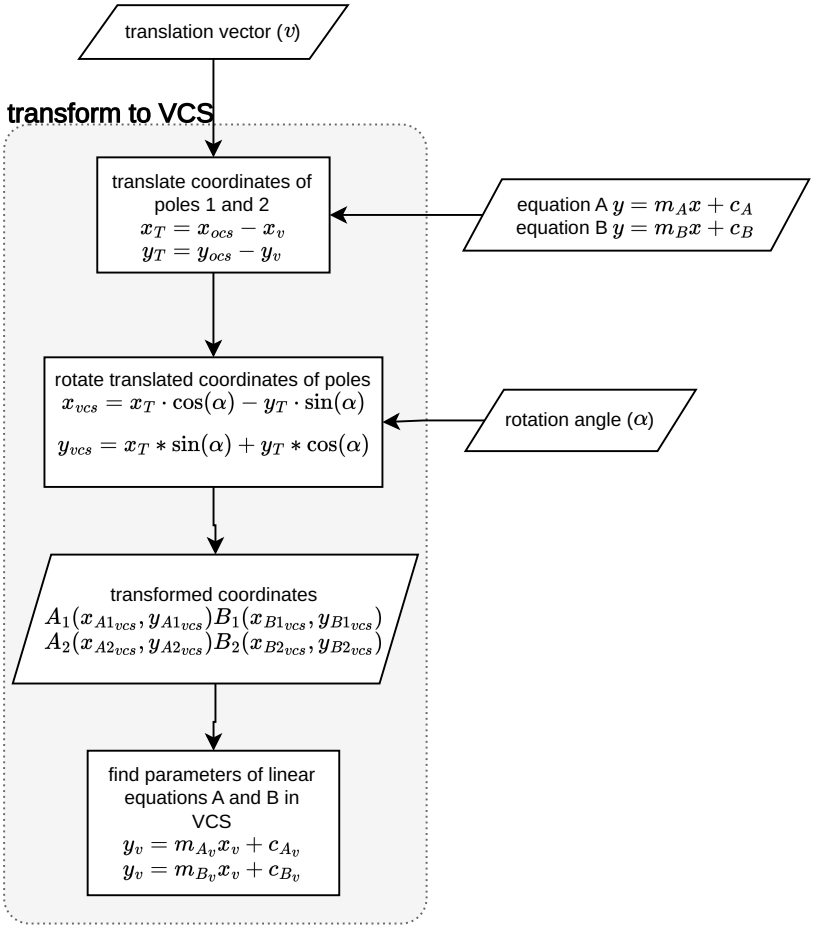

Figure 12: Flowchart for transformation to VCS. Step 3 of the program is to take user input coordinates for equations A and B and transform the values to the VCS.

7.2.4 B.4

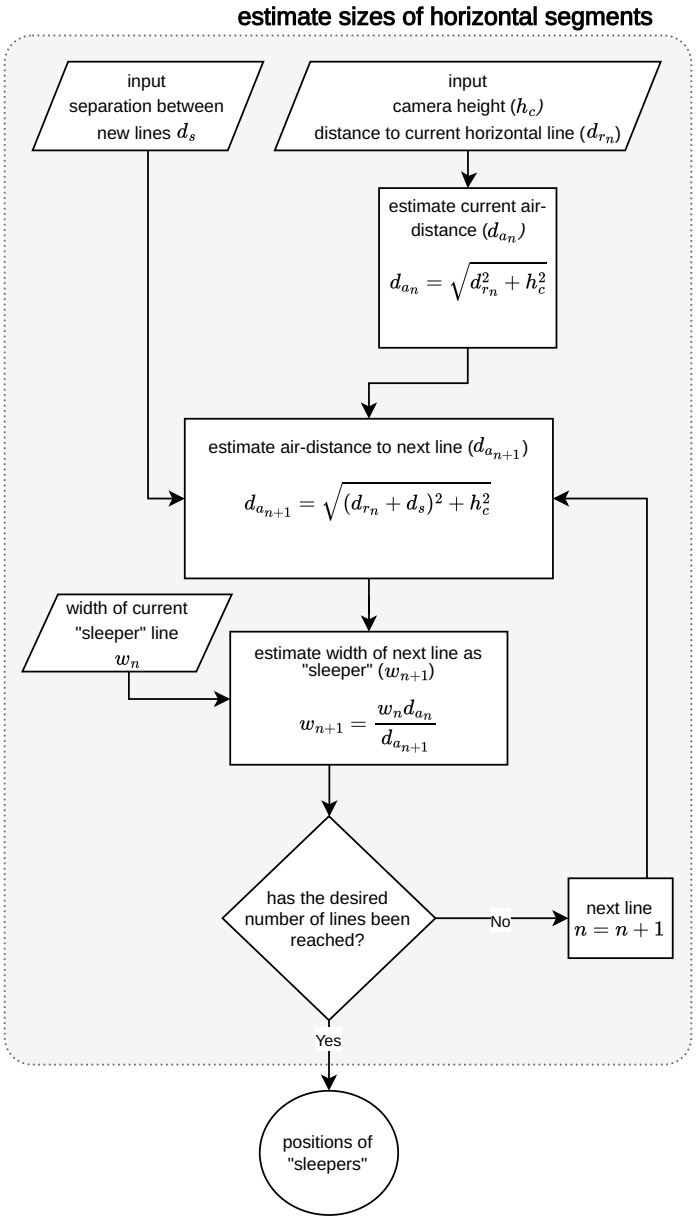

Figure 13: Step 4 of the program is to estimate the length of horizontal (parallel to VCS x-axis) lines that fit inside the perspective triangle.

7.2.5 B.5

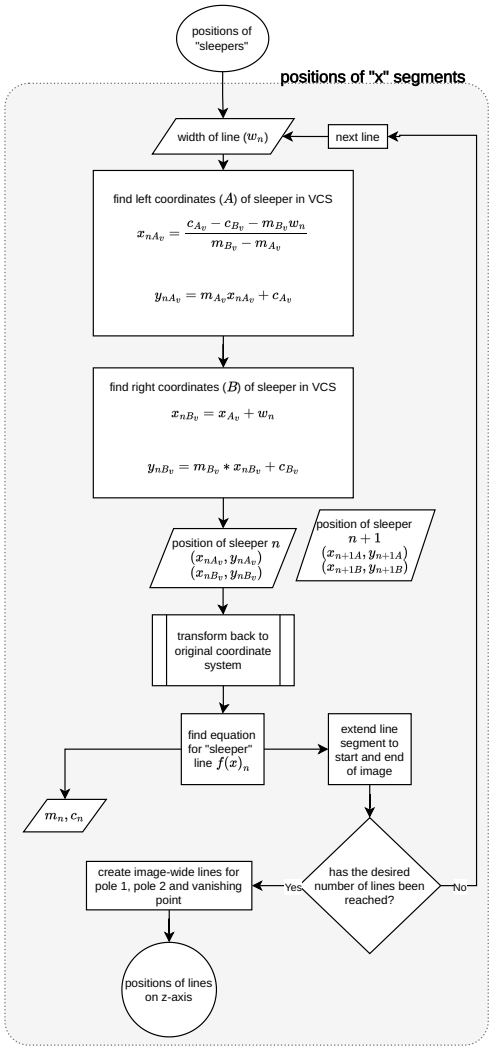

Figure 14: Step 5 of the program is to estimate the position of lines inside the perspective triangle and parallel to the x-axis.

7.2.6 B.6

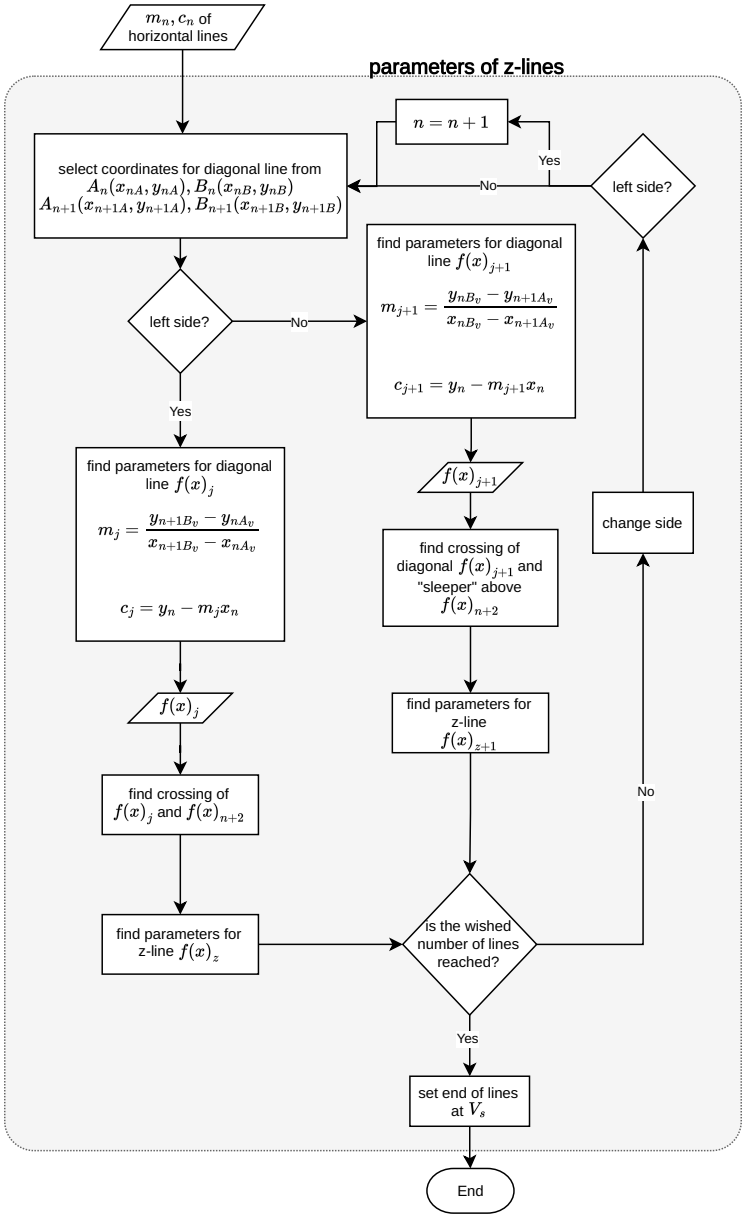

Figure 15: Step 6 of the program is find the parameters for lines that on the ground are perpendicular to the reference object.

7.3 Supporting Material C

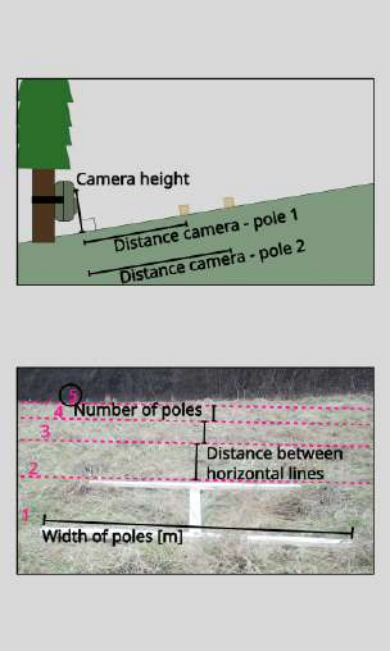

Figure 16: How to make measurements and how to input known distances.