**Proposed procedure for perspective transformation**

This section of the Supplementary Material describes the mathematical and coding procedure established to correct timelapse images and annotations for perspective, practically simplifying the procedure first introduced by Wakefield and Genin (1987). The code has been made available on GitHub: XXXX.

(1) Correction of angles of acceptance for water refraction

Angles of acceptance can be potentially corrected for water refraction using the Snell’s law, for example if the camera is enclosed in an air dome.

(1)

which becomes after readjustment,

(2)

Typical values for refractive indices are n1 (air) and n2 (seawater) are 1 and 1.33, respectively. Vertical and horizontal angles of acceptance will be injected as αair for refraction correction.

(2) Conversion of mid-edge coordinates

Let’s consider a situation where a camera is recording a flat seafloor with a given tilt angle at a given height (*h*) above the seabed. Vertical (*αv*) and horizontal angles (*αh*) of acceptance are known. *P* is the principal point, intersecting the optical axis with the image (Wakefield and Genin 1987).

A black background with a black square

AI-generated content may be incorrect.

Supplementary Figure 1. Schemes describing the position of parameters used in the demonstration originally set up by Wakefield and Genin (1987), from two different points of view: (A) lateral view and (B) tilt-perpendicular view from above.

From a lateral point of view (Supplementary Figure S1A), the complementary angle to the tilt angle can be found by:

(3)

Since the camera height vector (*h*) is perpendicular to the seabed, knowing the vertical acceptance angle of the camera, the tangent trigonometric function can be used to estimate the distance of the upper edge (*A*), the lower edge (*C*) and the principal point (*P*) of the image from the camera’s position over the seabed (Supplementary Figure S1A):

(4)

The length of the line of sight traveling from the principal point to the camera can be estimated using the cosine tangent function (Supplementary Figure 1A):

(5)

(6)

(7)

Looking at the camera from above, with an orientation equals to the tilt of the camera (Supplementary 1B), this orientation will allow a perpendicular view *D* which will conserve the same length as found in Equation 7. The distance between the lateral image edge and the principal point can be estimated (*E*).

(8)

The real-world coordinates of the points *a, b, c, d* within the image can be found using the distances derived with trigonometry (Supplementary Figure 1).

(9)

These points correspond to the middle of each edge of the original rectangular image. Considering the pixel width (*w*) and height (l) of the original image, their coordinates in the original image are:

(10)

(3) Computation of the homography matrix

Using the correspondence of this set of points, the transformation matrix can be computed using the get*PerspectiveTransform()* function of the OpenCV library (v.4.9.0.80; Python v.3.7). This function will derive a 3x3 transformation matrix *H* with 8 degrees of freedom. Note that the unit of the transformation will be inherited from the camera height above the seafloor.

To allow plotting the transformed image within the plot window, any residual offset will be removed to position the image on the vertical and horizontal axis. Since the transformed image will be an isosceles trapeze (Wakefield and Genin 1987), a horizontal offset (*L*) should be added to allow the image to fit within the plotting window. This horizontal offset can be adjusted iteratively based on the visual positioning of the transformed image.

(11)

(4) Application

For this study, parameters were the following:

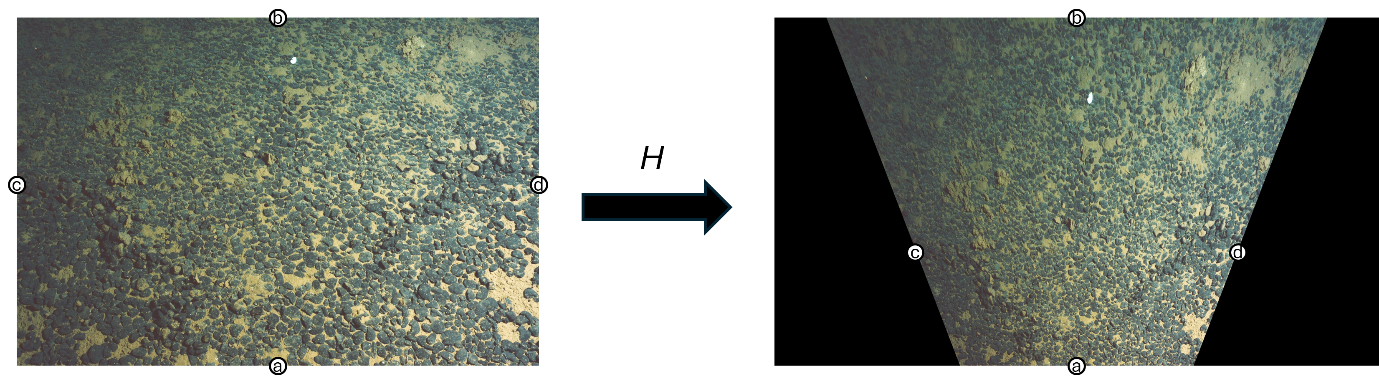
|  |  |
| --- | --- |
| *h* | 1450 mm |
| *Tilt angle* | 45° |
| *αh* | 57° |
| *αv* | 40° |
| *w* | 6000 pixels |
| *l* | 4000 pixels |
| *L* | 1000 mm |

Supplementary Table 1. Note corrected for seawater refraction

By encoding these parameters in the Python routine, the *getPerspectiveTransform()* determines the following transformation:

(12)

This transformation can be applied using the *warpPerspective()* function of the OpenCV library.



Supplementary Figure 1. Transformation of the original image using the homography matrix *H* (Equation 12) that was derived from the pixel and real-world coordinates of the four points indicated in this scheme (Equations 9-10).

The original image is now corrected for perspective. Each pixel in the corrected image equals one millimetre, since inherited by the unit of the camera height above the seafloor. The perspective transformation can be applied on annotations made on the original images to allow computing ecological metrics in a standard unit system. Furthermore, a regular grid of 100 mm can be overlaid over the perspective transformed image. The inverse of the homography matrix can be used to project the image back to the original pixel coordinate system.

A black and green grid with a black arrow

AI-generated content may be incorrect.

Supplementary Figure 2. Back transformation of the transformed image using the inverse of the homography matrix *H* (Equation 12). This procedure enables to rapidly draw a perspective grid for visualisation with the original image.

References:

Wakefield, W. W., and A. Genin. 1987. The use of a Canadian (perspective) grid in deep-sea photography. Deep Sea Research Part A. Oceanographic Research Papers **34**:469-478.