Automatically drawing vegetation classification maps using digital time-lapse cameras in alpine ecosystems

A Preprint

Ryotaro Okamoto *
Doctoral Program in Biology
University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki 305-8577 Japan.
okamoto.ryotaro.su@alumni.tsukuba.ac.jp

Hiroyuki Oguma
Biodiversity Division
National Institute for Environmental Studies
16-2 Onogawa, Tsukuba, Ibaraki 305-8506 Japan
oguma@nies.go.jp

Reiko Ide
Earth System Division
National Institute for Environmental Studies
16-2 Onogawa, Tsukuba, Ibaraki 305-8506 Japan
ide.reiko@nies.go.jp

November 24, 2022

Abstract

Alpine ecosystems are particularly vulnerable to the effect of climate change. Monitoring the distribution of alpine vegetation is required to plan practical conservation activities. However, conventional field observation and satellite remote sensing have difficulties in the coverage and frequency in alpine areas. Researchers have also utilized ground-based time-lapse cameras to observe alpine regions' snowmelt and vegetation phenology. Such time-lapse cameras have significant advantages in cost, resolution, and frequency. However, no research has used these to monitor vegetation distribution. This study proposes a novel method to draw georeferenced vegetation maps from ground-based imagery of the Japanese alps. Our approach has two components:classification and georectification methods. The proposed classification method uses pixel time series acquired from autumn images to utilize the patterns of autumn foliage. We show that using time-lapse imagery and a Recurrent Neural Network improves the performance of the vegetation classification. We also developed a novel method to transform a ground-based image into georeferenced data accurately. We propose 1. an automated procedure to acquire Ground Control Points (GCPs) and 2. a camera model that considers the lens distortion for accurate georectification. We show that our approach outperforms the conventional method. The proposed method achieved sufficient accuracy to observe the vegetation distribution on a plant-community scale. The evaluation reveals an F1 score of 0.937 in the vegetation classification and a root-mean-square error (RMSE) of 3.4 m in the georectification. Our results highlight the potential of cheap time-lapse cameras to monitor the distribution of alpine vegetation. The proposed method should significantly contribute to effective conservation planning of the alpine ecosystems.

***K*eywords** alpine ecology · deep learning · ecosystem monitoring

*https://github.com/0kam

1 Introduction

The effects of climate change on terrestrial ecosystems are particularly significant in alpine regions ([1]). Alpine vegetation depends on severe conditions such as low temperatures and long snow-covered periods. Thus alpine areas have unique species adapted to such extreme environments. Several studies have reported that recent global climatic changes, e.g., increasing temperatures and reducing snow-covered periods, have accelerated the invasion of non-native species into alpine areas (see [2]). In Japan, dwarf bamboo (*Sasa kurilensis*) has invaded alpine snow meadows, probably driven by the extension of the snow-free period ([3]). Climate change has also affected the growth and phenologies of native species. For example, the growth of dwarf pine (*Pinus pumila*), a dominant species in Japanese alpine regions, have been affected by climatic conditions such as temperature and snowmelt ([4]). In addition, researchers have reported that such changes vary depending on species and microhabitats ([5]). Understanding and predicting the undergoing changes of alpine vegetation requires a monitoring method scalable to a wide range with a spatially and temporally high resolution.

Previous studies have mainly depended on field observations, yet it is hard to cover broad areas in alpine regions due to the poor accessibility and severe weather. Satellite, airborne, and Unmanned Aerial Vehicle (UAV) remote sensing methods are alternatives. However, satellite imagery of alpine areas is rarely available due to cloud cover, and the spatial resolution is not enough to observe vegetation changes at the plant community scale. Airborne imagery can obtain high-resolution data, but its cost becomes a bottleneck for frequent monitoring. Although UAV methods have become a cost-effective tool for ecological monitoring ([6]), operating UAVs in alpine regions is sometimes challenging due to the strong wind and harsh topology.

On the other hand, researchers have also utilized automated digital time-lapse cameras mounted on the ground for monitoring green-leaf phenologies in forests ([7]), grasslands ([8]), and alpine meadows ([9]). Unlike satellite imagery, ground-based cameras provide images free of clouds and atmospheric effects. Also, they can obtain high-resolution (i.e., sub-meter scale) and frequent (i.e., daily or hourly) images at a meager cost. Most previous studies with such cameras were about calculating the phenology index (e.g., excess greenness, [10]) by setting some regions of interest (ROI) in the images. Few studies have utilized such images in monitoring vegetation distributions. This lack of studies seems to be because applying such time-lapse imagery in monitoring vegetation distribution has two technical challenges. First, unlike satellites' multispectral sensors, ordinal digital cameras can only obtain three bands (Red, Green, and Blue), making it harder to classify the vegetation. Second, since digital time-lapse cameras are mounted on the ground, transforming these images into geospatial data (e.g., orthoimage, this process is called georectification) is challenging. In other words, even if we classify the vegetation from such images, we cannot quantitatively measure and analyze it using Geographic Information Systems (GIS) tools. Georectification is essential for utilizing time-lapse cameras in scientific conservation planning.

This study proposes an automated method for drawing vegetation classification maps with a digital time-lapse camera by solving these two challenges. We solved the first challenge by using time-lapse images to classify vegetation. The autumn leaf phenology, which varies among species, was utilized in the classification process. We show the effectiveness of such phenological information in vegetation classification. A novel georectification method for ground-based photographs was also developed to tackle the second challenge. Using the proposed method, we generated a vegetation classification map from time-lapse images taken in the Japanese alps. Our approach enables us to continuously monitor vegetation distribution and its changes at a plant-community scale, covering broad areas at a meager cost.

2 Materials and methods

The proposed method has three steps: 1. Image-to-image alignment for correcting the movements of the camera. 2. Vegetation classification for classifying each pixel into vegetation categories. 3. Georectification for transforming vegetation classification results into geospatial data. Fig. 1 shows the overview of the entire process. The proposed method requires training data for the vegetation classification model, a Digital Elevation Model (DEM), and a georectified airborne/satellite image that covers the camera's field of view.

Fig. 1 shows the overview of the entire process. The proposed method requires a teacher dataset for training the vegetation classification model, a Digital Elevation Model (DEM), and a georectified airborne/satellite image that covers the camera's field of view for the georectification process.

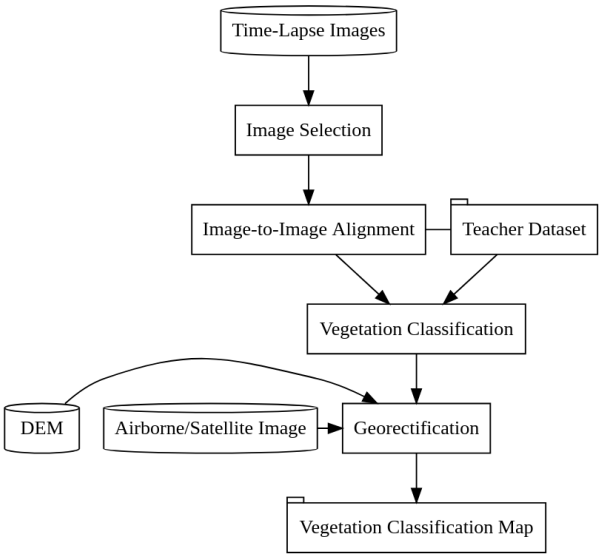

Figure 1: An overview of the proposed procedure

2.1 Time-lapse image acquisition

Our approach is intended for digital time-lapse images of alpine landscapes. In this study, we demonstrated it with time-lapse photos of a Japanese alpine region owned by the National Institute for Environmental Studies, Japan (NIES). All the images are publically available on NIES' webpage (`https://db.cger.nies.go.jp/gem/ja/mountain/station.html?id=2`). In 2010, NIES installed the digital time-lapse camera (EOS 5D MK2, Canon Inc., 21 M pixels) on a mountain lodge Murodo-sanso (about 2350 m a.s.l., above the forest limit), located at the foot of Mt. Tateyama (3015 m a.s.l.), in the Nothern Japanese Alps (Fig. 2). The photographs were taken hourly, from 6 a.m. to 7 p.m. The camera's field of view (FoV) includes Mt. Tateyama, which ranges from about 2350 m a.s.l. to 3015 m a.s.l. in elevation. The pixel size on the ground was about 0.5 m at the ridge, about 1 km from the cameras. The area has a complex mosaic-like vegetation structure because of its topography, including rocks, cliffs, curls, and moraines. From April to November, the camera observed the snowmelt and seasonal phenology of evergreen(e.g., *Pinus pumila*) and deciduous (*Sorbus* sp., *Betula ermanii*) dwarf trees, dwarf bamboos (e.g., *Sasa kurilensis*), and alpine shrubs and herbaceous plants (e.g., *Geum pentapetalum*, *Nephrophyllidium crista-galli*).

2.2 Preprocessing

2.2.1 Image selection

In the beginning, we selected seven images with good weather from late summer to late fall (25th Aug, 5th, 12th, 20th, 26th Sep, 3rd, and 10th Oct) in 2015. Since the temporal patterns of autumn foliage coloration vary among species (Fig. 3), this season is suitable for classifying vegetation. While acquiring a series of images costs a lot with UAVs and airborne imagery, time-lapse cameras are particularly good at observing such temporal patterns of leaf color.

2.2.2 Automatic image-to-image alignment

Time-lapse images often suffer from the camera's movement caused by strong wind, human interference, and thermal expansion of the camera platform. Since such movement can cause errors in vegetation classification and georectification, we implemented an alignment method that aligns each image to one image called the master image. The software was developed using Python3 programming language and OpenCV4 (`https://opencv.org/`) image processing library. We employed a feature-based alignment method used by previous studies ([11], [12]). First, we searched and matched key points between an image and the master image using the AKAZE local feature extractor ([13]) and K-nearest neighbor matcher. Then, we searched and applied the homography matrix that minimizes the distance between each pair of the matching points,

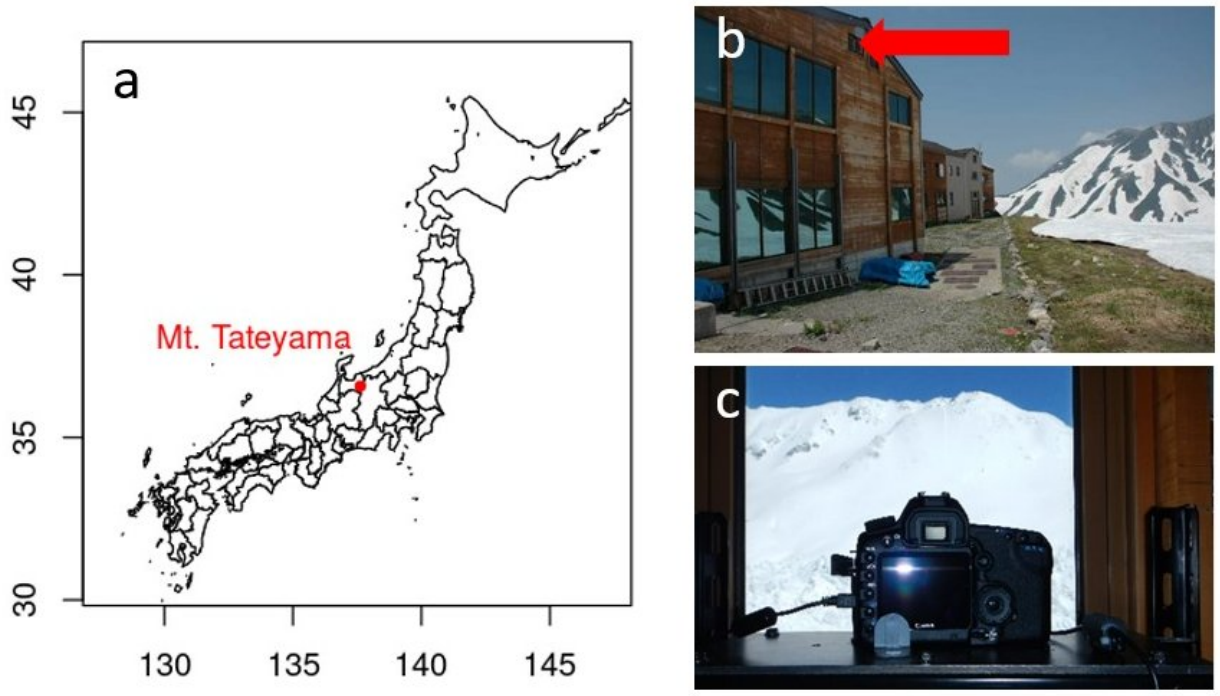

Figure 2: The study site. (a) The location of Mt. Tateyama. (b) The installation point of the camera in Murodo-sanso lodge. (c) The camera (EOS 5D MK2, Canon Inc.)

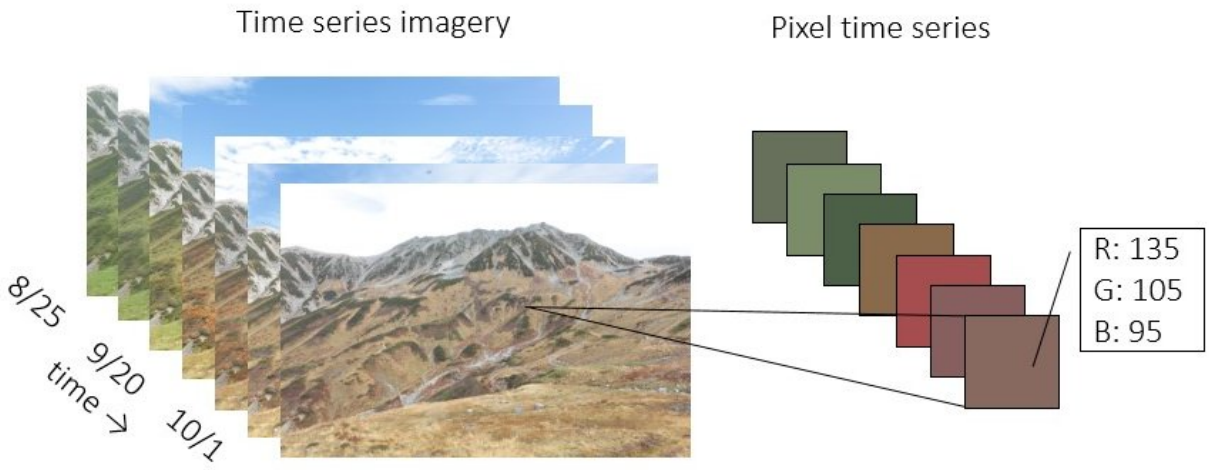

Figure 3: Pixel time series acquired from the time-lapse camera. You can obtain a time series of pixel values (i.e., Red, Green, and Blue) for each pixel on the photographs (left). Such pixel time series reflects the autumn phenology of the vegetation and varies among vegetation classes (right).

using OpenCV's "findHomography" function (2.2.2). Applying the estimated homography matrix, we could align the images accurately (0.654 pixels in root mean square error (RMSE) of the matching points). Finally, an input mask was applied to define image regions that should be ignored in the following procedure, such as the sky and the regions too close to the camera. This process ensures that a set of pixel time series reflects phenological information of the same location, which is essential in the vegetation classification process.

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = H \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{1}$$

2.3 Vegetation classification

After the image-to-image alignment, we stuck the images and extract a time series of pixel values (Red, Green, and Blue, see Fig. 3) for each pixel. Such pixel time series reflects the temporal patterns of leaf colors. Because deciduous plants have great differences in autumn leaf phenology among species, researchers have used that information for vegetation classification with satellite imagery ([14], [15], [16]). However, no research has applied this technique to ground-based time-lapse imagery. We implemented Support Vector Machine (SVM) and Recurrent Neural Network (RNN) based vegetation classifiers and tested the effects of using pixel time series of the ground-based imagery on the classification performance.

2.3.1 Model Architecture

We prepared two supervised models, SVM and RNN, to classify the pixel time series into vegetation categories. SVM is one of the most popular machine-learning models in remote sensing and has many applications ([17]), including vegetation classification with multitemporal satellite imagery ([15]). RNN is a neural network that recurrently processes sequential or temporal steps of data to recognize their dynamics. RNN has been used in many tasks such as speech recognition ([18], [19], [20]), and machine translation ([21], [22]). Researchers have also utilized RNN for remote sensing tasks, such as land cover classification, with multi-temporal satellite imagery ([23], [24]). ([23]) reported that an RNN classifier outperforms an SVM classifier on the land cover classification accuracy. Amongst many variants of RNN, we used Long Short-Time Memory (LSTM, [25], one of the most well-known RNN architectures. Batch-normalization technique ([26]) and the GELU activation function ([27]) were employed to speed up and stabilize the model training. As a comparison, we also classified the pixel of every single image separately using SVM classifiers to test whether using multi-temporal imagery improves the classification performance.

2.3.2 Dataset preparation

Each pixel time series was classified into 7 vegetation classes: Dwarf Pine (*Pinus pumila*), Dwarf Bamboo (*Sasa kurilensis*), Rowans (*Sorbus sambucifolia*, *Sorbus matsumurana*), Maple (*Acer tschonoskii*), Montane Alder (*Alnus viridis* subsp. *maximowiczii*), Other Vegetations (such as alpine shrubs and herbaceous plants), and No Vegetation. Experts prepared a teacher dataset for each class using an open-source image annotation software (Semantic Segmentation Editor, Hitachi, `https://github.com/Hitachi-Automotive-And-Industry-Lab/semantic-segmentation-editor`). The prepared teacher dataset contains XX pixels in XX polygons, which covers about X% of the image. Each polygon was validated with telephotos taken at the same place in 2016. For some polygons that were not seen clearly even in the telephoto, we conducted a field survey in the November of 2022.

2.3.3 Experimental Settings

We implemented the classifiers with Python3 language. For the SVM classifier, we used the ThunderSVM library (`https://github.com/Xtra-Computing/thundersvm`) and an RBF kernel with a regularization parameter equal to 1.0. The gamma value was set automatically by the "scaled" setting of the ThunderSVM library as $\gamma = \frac{1}{n_{features} \times V(X)}$, where and stand for the dimensions and variance of the input data respectively. The RNN classifier was developed using the PyTorch deep neural network library (`https://pytorch.org/`). We set the number of hidden dimensions equal to 5. We trained the model 200 epochs with the RAdam optimizer ([28]) with a learning rate of 0.001 and a batch size of 500. To validate the performance of the different classifiers, we performed a 5-fold cross-validation. Each fold was stratified with the vegetation categories. As a model evaluation metric, we used the per-class F1-score, a standard metric in machine

learning (2.3.3). TP, FP, FN stand for the numbers of true positives, false positives, and false negatives, respectively. All source codes are available via GitHub (`https://github.com/0kam/VegetationMapPaper`).

$$\begin{aligned} precision &= \frac{TP}{TP + FP} \\ recall &= \frac{TP}{TP + FN} \\ F_1 &= \frac{2 \cdot precision \cdot recall}{precision + recall} \end{aligned} \tag{2}$$

2.4 Automatic georectification

The second challenge of utilizing time-lapse cameras in vegetation monitoring is the difficulty of georectification, i.e., georeferencing and transforming the images into geographic data. We developed a novel method to perform this process automatically and accurately. Georectification of ground-based images has been a difficult task, and this causes the underuse of potentially rich information in ground-based imagery. In plain words, georectification means aligning images onto Digital Surface Models (DSMs) so that every pixel of an image gets a geographical coordinate. It usually requires a camera model. A camera is considered a function that transforms 3D geographical coordinates (e.g., X, Y, and height in a Universal Transverse Mercator coordinate system (UTM)) into 2D image coordinates (locations of each pixel in the image). Estimating the parameters of this function (such as the camera location, pose, and the FoV), you can simulate how each point of the DSM appears in the image. Usually, georectification has three steps:

1. Finding Ground Control Points (GCPs) in the image.
2. Estimating the camera parameters such as the camera pose and FoV using GCPs.
3. Projecting the image onto the DSM using the camera parameters.

Recently, researchers have developed some georectification methods to use ground-based photographs in glaciology ([29], [11]) and snow cover studies ([12]). Usually, GNSS-positioned GCPs (such as GCP markers) are set before photo acquisition to acquire GCPs. However, setting such markers in alpine areas is often difficult due to their rugged topology. ([12]) tackled this problem by developing a method that uses mountain silhouettes to match images and DSMs directly. However, this silhouette-based method has a drawback in projection accuracy. It ignores lens distortion and only uses limited areas (silhouettes) of images in the GCP acquisition. To overcome these challenges, This study proposes 1. a camera model which includes lens distortion and 2. a novel method to acquire GCPs in a broader area than the silhouette-based method.

2.5 Modeling and estimating camera parameters

In this section, we explain our camera model with lens distortion modeling. As mentioned, we modeled a camera as a function that transforms geographical coordinates into image coordinates. We implemented this procedure using the OpenGL framework (`https://www.opengl.org`) and the ModernGL Python library ([30]) to accelerate it with graphical processing units (GPUs). In OpenGL, we can divide this process into three operations:

1. Transforming the geographical coordinates (also called the world coordinates) to the view coordinates (coordinates seen from the camera's point of view) using the camera's extrinsic parameters (i.e., the location and angles of the camera).
2. Distorting the view coordinates using the lens distortion parameters.
3. Transforming the view coordinates to the image coordinates (also called the screen coordinates) using the camera's intrinsic parameters (i.e., the camera's FoV).

First, we transformed the geographic coordinates into the view coordinates, which are relative to the camera's position and direction. We used a 4×4 view matrix (3) M_{view} in this step. The view matrix represents the camera's position and direction (pan, tilt, roll).

$$M_{view} = \begin{bmatrix} \cos roll & -\sin roll & 0 & 0 \\ \sin roll & \cos roll & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos tilt & -\sin tilt & 0 \\ 0 & \sin tilt & \cos tilt & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos pan & 0 & \sin pan & 0 \\ 0 & 1 & 0 & 0 \\ -\sin pan & 0 & \cos pan & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -z \\ 0 & 0 & 1 & -y \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Where $pan, tilt, roll$ are the Euler angles of the camera pose and x, y, z are the camera location in the geographic coordinate system. Applying the view matrix M_{view} transforms the geographic coordinates $[X_{geo} \quad Z_{geo} \quad Y_{geo} \quad 1]$ (the horizontal, vertical, and depth coordinates, respectively) into the view coordinates $[X_{view} \quad Z_{view} \quad Y_{view} \quad 1]$ (4). In the OpenGL's view coordinate system, X_{view}, Z_{view} and Y_{view} represents horizontal, vertical, and depth positions respectively. Note that the geographic coordinate system must be cartesian (such as the UTM coordinate systems, not the lat/long coordinate system).

$$\begin{bmatrix} X_{view} \\ Z_{view} \\ Y_{view} \\ 1 \end{bmatrix} = M_{view} \begin{bmatrix} X_{geo} \\ Z_{geo} \\ Y_{geo} \\ 1 \end{bmatrix} \tag{4}$$

Second, we distorted the view coordinates to simulate the lens distortion. We modeled the lens distortion (5) based on [31] and OpenCV's implementation (`https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html`), where X_{norm} and Z_{norm} are the Y-normalized view coordinates. This model distorts points' locations according to the distance from the center of the image when projected to the image surface. Our model includes radial ($k1$~$k6$), tangental ($p1$, $p2$), thin prism ($s1$~$s4$) distortion, and unequal pixel aspect ratio ($a1$, $a2$). See [31] for the details of lens distortion modeling.

$$\begin{aligned} X_{norm} &= \frac{X_{view}}{Y_{view}} \\ Z_{norm} &= \frac{Z_{view}}{Y_{view}} \\ r^2 &= {X_{norm}}^2 + {Z_{norm}}^2 \\ \begin{bmatrix} X_{dist\_norm} \\ Z_{dist\_norm} \end{bmatrix} &= \begin{bmatrix} X_{norm} \frac{1+k_1 r^2+k_2 r^4+k_3 r^6}{1+k_4 r^2+k_5 r^4+k_6 r^6} + 2p_1 x'y' + p_2(r^2+2x'^2) + s_1 r^2 + s_2 r^4 \\ Z_{norm} \frac{1+a_1+k_1 r^2+k_2 r^4+k_3 r^6}{1+a_2+k_4 r^2+k_5 r^4+k_6 r^6} + p_1(r^2+2y'^2) + 2p_2 x'y' + s_3 r^2 + s_4 r^4 \end{bmatrix} \\ X_{dist} &= X_{dist\_norm} Y_{view} \\ Z_{dist} &= Z_{dist\_norm} Y_{view} \end{aligned} \tag{5}$$

Lastly, we transformed the distorted view coordinates into the image coordinates by perspective projection. During the perspective projection, a closer object is drawn larger. We used a 4×4 projection matrix M_{proj} (6), that represents the camera's horizontal (fov_x) and vertical (fov_z) FoV. Finally, we got the image coordinates as X_{image}, Z_{image} (7).

$$\begin{aligned} f_x &= \frac{1}{\frac{\tan fov_x}{2}} \\ f_z &= \frac{1}{\frac{\tan fov_z}{2}} \\ M_{proj} &= \begin{bmatrix} f_x & 0 & 1 & 0 \\ 0 & f_z & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \end{aligned} \tag{6}$$

$$\begin{bmatrix} X_{image} \\ Z_{image} \\ Y_{image} \\ 1 \end{bmatrix} = M_{proj} \begin{bmatrix} X_{dist} \\ Z_{dist} \\ Y_{dist} \\ 1 \end{bmatrix} \tag{7}$$

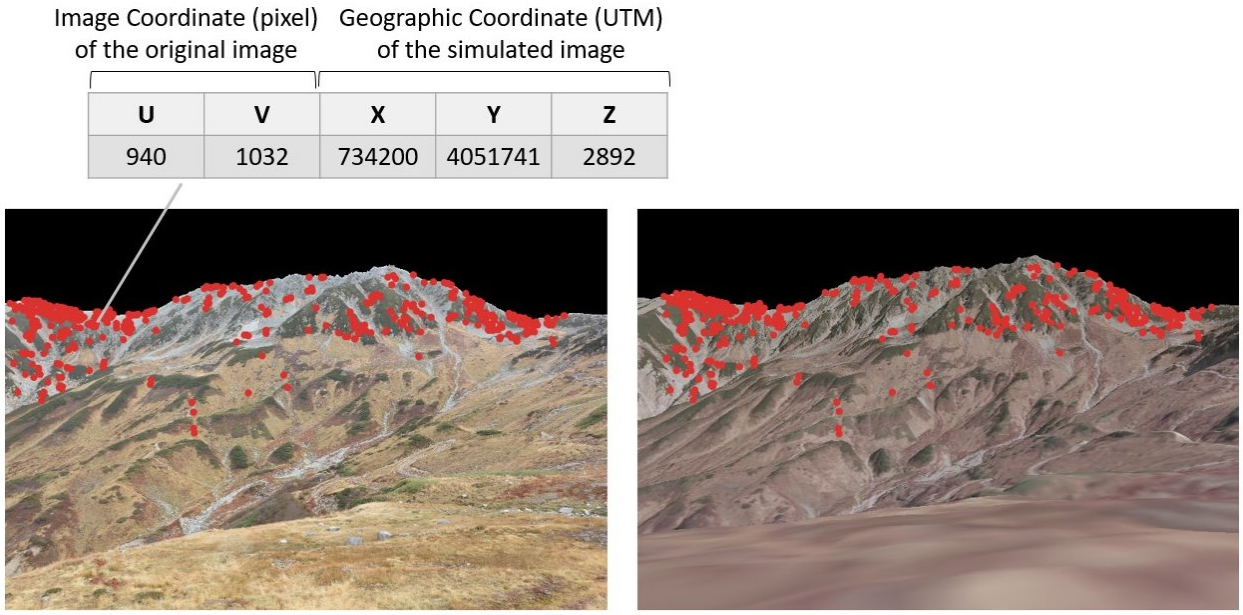

Figure 4: Automatic acquisition of GCPs. The original image (left) and the simulated image (right). The simulated image was rendered with an orthophoto, DSM, and initial camera parameters. Red points show the matching points. We used these points as GCPs.

2.6 Image-matching-based aqcuisition of GCPs

The second weak point of the previous silhouette-based method is a low georectification accuracy on the mountainside. We developed a novel image-matching-based method to acquire GCPs in a broader area. Before processing, an orthorectified airborne/satellite image (orthophoto) and a DSM that covers the camera's field of view, and a set of initial camera parameters (roughly estimated camera parameters) should be prepared. Our method consists of two parts: 1. rendering a simulated landscape image and 2. matching the simulated and original image. In the first step, we rendered a simulated landscape image by applying the camera model to the DSM and orthophoto. Every pixel of this simulated image has a geographic coordinate. In the second step, we searched for matching key points between the original image and the simulated image (4) by the same AKAZE feature-based method used in the image-to-image-alignment process. Since these matching points have both geographical coordinates (from the DSM) and image coordinates (from the original image), we used them as GCPs. This method enables us to acquire GCPs in a much broader area than the silhouette-based method. Note that the quality of the DSM and the orthorectification accuracy of the orthophoto may affect the accuracy of GCPs.

2.7 Camera parameter estimation

The acquired GCPs were used to estimate the camera parameters of the image. Applying the camera model to the GCPs' geographical coordinates, we can calculate the reprojected image coordinates with a set of camera parameters (the camera location, pose, FoV, and lens distortion parameters). Then, we can measure the distance from the actual image coordinates of GCPs (U, V of Fig. 5) to the reprojected image coordinates (U', V' of Fig. 5). We estimated the camera parameters (except camera location) by minimizing this distance using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES, ([32]), Fig. 5), which performs well in optimizing a multimodal, complex function with less parameters to be set. We could not estimate the camera location because it makes the problem too complicated (e.g., a telephoto taken from a distance and a wide-angle taken from a close look similar).

2.8 Georectification of the vegetation map

At last, we georectified the vegetation classification result using the optimized camera parameters (Fig. 6). At this point, we got the point data of vegetation (as shown in the table on the right of Fig. 6) which every

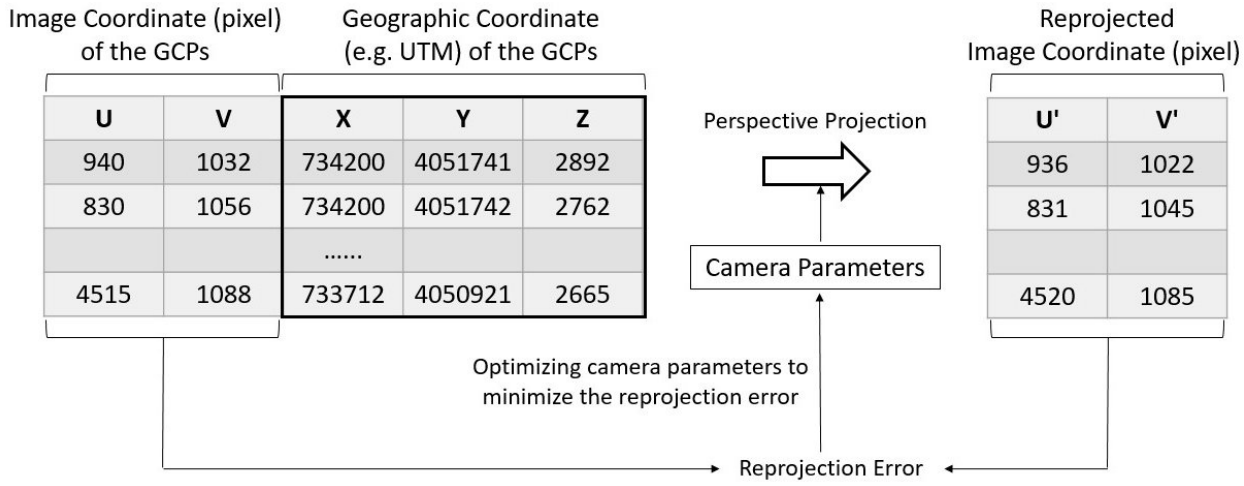

Figure 5: Workflow of the camera parameter optimization. We estimated the camera parameters by minimizing the GCP's reprojection error.

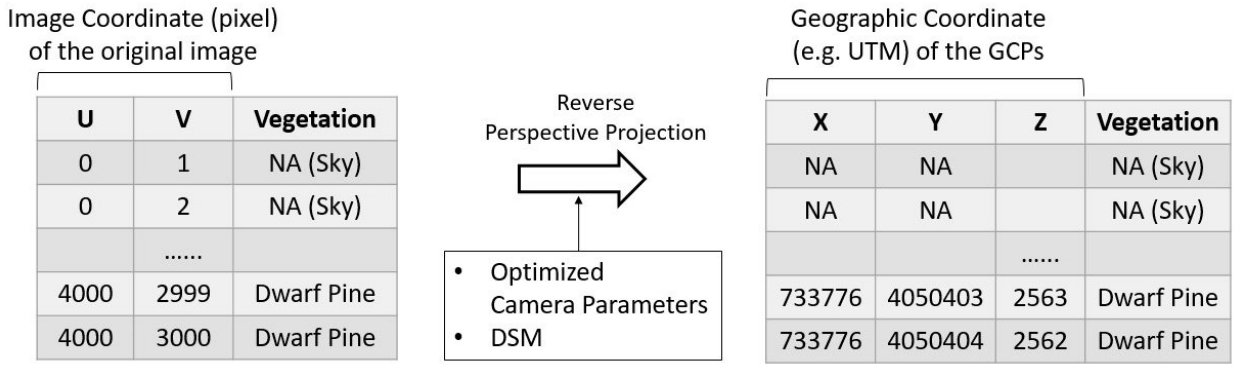

Figure 6: Our georectification procedure. We applied the optimized camera parameters to the DSM and the vegetation classification result to get a vegetation classification map.

row represents a pixel of the original image. To measure the area of each vegetation class, we converted the point data to raster data. We rasterized the point data in 1 m resolution, then interpolated holes up to 2 m since the maximum pixel footprint was about 2 m on the ridge of the mountain. This rasterization procedure was implemented using the R language (R Core Team, 2022), the "stars" package (Pebesma, 2022), and the "terra" (J Hijmans, 2022) package. See `https://github.com/0kam/VegetationMapPaper/blob/master/scripts/utils/interpolate.R` for the source code.

2.9 Implementation and data set

We implemented the procedure with Python3 language and published it as an open-source package via GitHub (`https://github.com/0kam/alproj`). We used an airborne image taken in October 2014 with a spatial resolution of 1.0 m. Also, we used the 1m resolution Digital Surface Model that was also used in the orthorectification process of the airborne image.

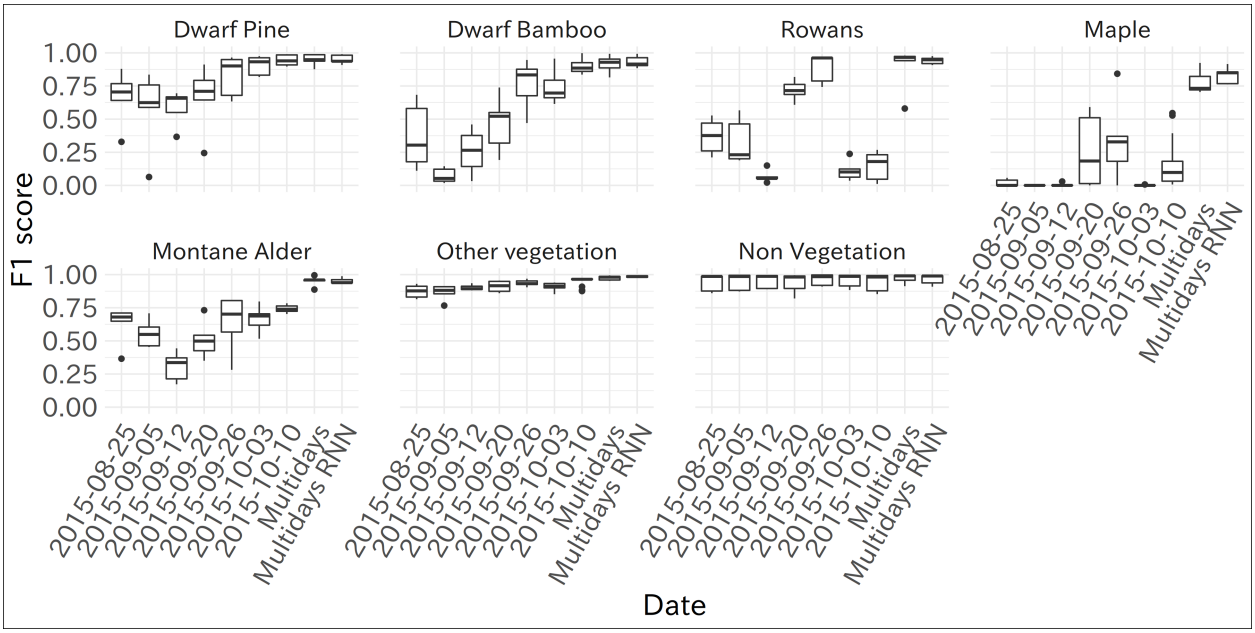

Figure 7: F1-scores of the vegetation classification. F1-score was calculated for each vegetation class. Each box plot shows the results of a 5-fold cross-validation. We classified each pixel of single image (shown as the shooting date) with SVM classifiers and pixel time-seriese with SVM (shown as "Multidays") and RNN (shown as "Multidays RNN") classifiers.

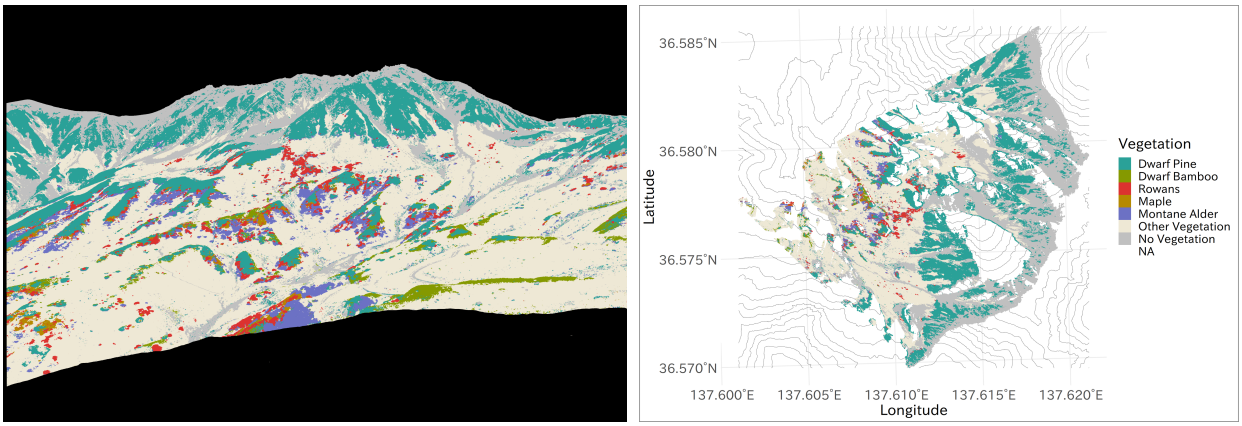

Figure 8: Left: Vegetation classification results of the RNN model. The masked area, i.e., the sky and the regions too close to the camera, is shown in black. Right: The generated vegetation map. The background shows contour lines.

3 Results

3.1 Vegetation classification accuracy

Fig. 7 shows the results of the 5-fold cross-validation. Focusing on the single pixel classification (shown as the shooting date), the best date for classification was different among classes. For example, 26th Sep was the best for identifying Rowans, but 10th Oct was for Dwarf Pine and Dwarf Bamboo. The autumn colorization of Rowans was at best on 26th Sep, and most vegetation except Dwarf Bamboo and Dwarf Pine were blasted on 10th Oct. No single image can classify Maple and Montane Alder accurately. In contrast, using all images (shown as "Multidays" and "Multidays RNN" representing the SVM and RNN classifiers, respectively) resulted in a high F1 score in every class. Also, the RNN classifier outperforms the SVM classifier (0.937 and 0.918 in macro average F1 score). Fig. 8 (left) shows the products of the RNN classifier. We can observe the distribution of each vegetation class at the plant-community scale.

3.2 Georectification accuracy

We tested the accuracy of our georectification method and evaluated the effect of its two features: using the simulated image and image matching to acquire GCPs and modeling the lens distortion. We manually searched 83 matching points between the simulated and original images and treated them as the test GCPs: GCPs that are not used in the georectification process. We additionally prepared two comparison methods to test the performance of the proposed method. The first one is the silhouette-based matching method (after called the silhouette method) used in previous studies ([12]). The second is the proposed method without the lens distortion model (after this called no distortion). We evaluated the projection error of the test GCPs using these three methods. The proposed method achieves accurate projection (3.45 m as the root mean square error) while the other two did not (16.1 m with silhouette, 23.6 m with no distortion). Note that while (Portenier et al., 2020) does not consider lens distortion, we did in the silhouette method. To investigate the tendency of our method's georectification error, we also checked the relationship between the projection error and the distance from the shooting point for each test GCP (Fig. 9). The projection error was especially large in the GCPs near the shooting point. Fig. 8 (right) shows the produced vegetation map. The vegetation map has missing areas because ridges blocked the camera's view.

4 Discussion

We suggested an automated procedure to transform time-lapse images of an alpine region into a georeferenced vegetation map at a meager cost. This task is challenging because of 1. the difficulty of classifying vegetation with ordinal digital camera imagery and 2. the difficulty of georectification of such ground-based imagery. We solved these issues by 1. using the temporal information of autumn leaf colors for vegetation classification and 2. developing a novel method for accurate image georectification. The proposed methods reached a functional performance (0.937 in a mean F1 Score of vegetation classification, 3.4 m in a mean projection error of georectification) to use the vegetation map in monitoring the vegetation distributions on a plant-community scale.

4.1 The benefit of using time-series imagery for vegetation classification

One of the shortcomings of ordinal digital cameras is that they only have three bands (Red, Blue, and Green). We made up for this lack of information using the rich temporal information that time-lapse cameras can obtain. Many plant species have characteristic phenologies, such as autumn foliage, and the suitable season to observe them differs among species. The accuracy of the single-pixel classification (shown in the shooting date in Fig. 7) shows that each vegetation class has suitable timing to identify it in photographs. Classifying multiple vegetation classes requires a long-term (such as several months long) and frequent (such as daily) monitoring, and digital time-lapse cameras are suitable. In addition, combining multi-temporal images improved the classification accuracy, especially with Maple and Montane Alder. The best F1 score was low with single pixel classification (0.750 for Montane Alder, 0.345 for Maple), and using multi-temporal images improves them much (0.951 and 0.781 with the SVM classifier). This result shows that identifying these classes requires multi-temporal information. The inter-community heterogeneity of autumn foliage timing in these species may cause this result. As the previous study pointed out ([23]), the RNN model outperformed the SVM model, particularly in identifying Maple (0.781 with SVM and 0.831 with RNN). Our model can still be further improved. As ([24]) claimed, the patch-based classification method may improve

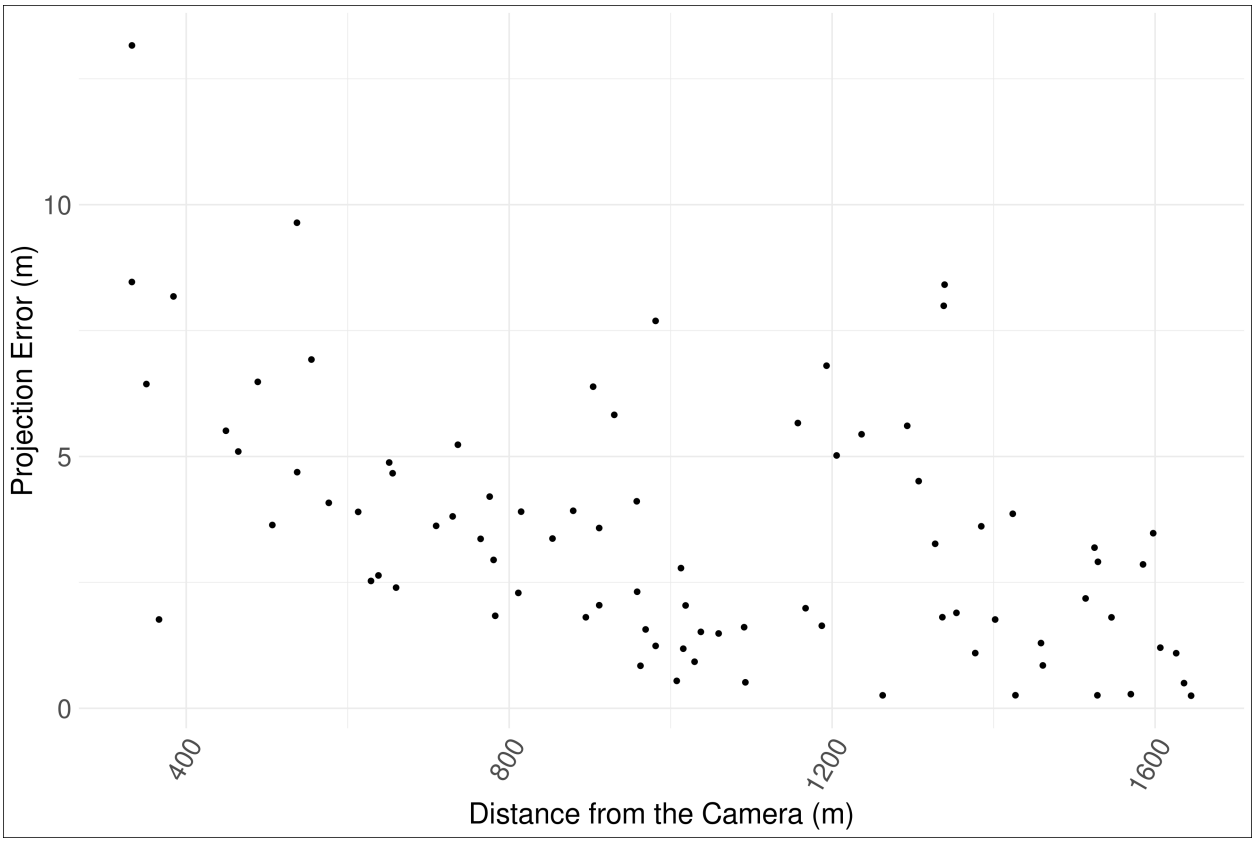

Figure 9: The projection error of the proposed method. The projection error was particularly large in the area near the camera.

the performance. Such patch-based models take a time series of small patches as an input to utilize the spatial information, such as textures, in the classification process.

4.2 The performance of the georectification method and its limitation.

The most significant barrier to using ground-based imagery in ecological monitoring is its difficulty in treating it as geospatial data. An automated and accurate georectification method was required to overcome this challenge. Without georectification, we cannot quantitatively analyze (e.g., measuring the area, locating the place, and analyzing with topographic data) the acquired information, such as the vegetation distribution. We suggested a novel method to georectify such imagery. With minimal manual user inputs (i.e., the location of the camera and the initial camera parameters), our method achieved a state-of-the-art accuracy (3.45 m). We tested the effectiveness of the two features of our method: the lens distortion model and the image-matching-based method to acquire GCPs automatically. The results show that the lens distortion model greatly improved the georectification accuracy. Without the lens distortion model, our model resulted in a projection accuracy of 23.6 m. However, since the effect of lens distortion on the georectification accuracy is different among cameras and lenses ([12]), further testing will be required to ensure the advantage of our lens distortion model. Note that the proposed lens distortion model does not support a fisheye lens and omnidirectional camera. Another feature of our method is the automatic acquisition of GCPs using simulated images and image-matching techniques. This method requires a high-resolution orthophoto that covers the camera's field of view. This costs a bit but the reward is considerable. Using simulated images in GCP acquisition also improved the georectification accuracy; the conventional silhouette-based method resulted in an RMSE of 16.1 m. This result suggests that acquiring matching points between the image and the DSM in a broader area benefits the camera parameter estimation process. Our method assumes that the landscape (e.g., topography and vegetation cover) looks the same between the original image and the simulated image. While we employed a robust matching method, large changes in mountain terrain (such as landslides) may

cause a matching error. Slight changes in vegetation cover will also affect the quality of GCPs. However, the negative effect of it (a few meters) should be much smaller than the positive effect (about 13 m) of using these points as GCPs. The proposed method (AKAZE image matcher) could not acquire matching points in the area near the camera, and the projection error was relatively large (Fig. 9). While the spatial resolution of the simulated image depends on orthophoto (in this study, 1 m), that of the original image changes depending on the distance from the camera. In such near areas, the spatial resolution of the original image is much higher than that of the simulated image, and the image matcher cannot handle the difference. Recently, researchers have developed some deep-learning-based image-matching methods (e.g., [33], [34]) that are more robust to the differences in image sources and viewpoints. Applying these methods to the GCP acquiring process may improve the performance of our method, especially in near areas. Theoretically, our approach can handle the images of other ecosystems, such as forests. However, because the georectification accuracy strongly depends on the accuracy of the DSM, this may be difficult. Since alpine plants tend to be very short, readily available Digital Elevation Models (DEMs) can be used as the DSMs. In forestry areas, vegetation height is considerable, and the gap between the DEMs and the actual surface (top of the forest) will cause significant errors in the georectification process.

4.3 Future application and conclusion

Here, we proposed an automated method to draw a vegetation map from a series of images acquired by a digital time-lapse camera. The produced vegetation map has sufficient accuracy for monitoring endemic and vulnerable alpine vegetation. One of the benefits of digital time-lapse cameras is their long-term operability. Applying the proposed method to the existing long-term time-lapse imagery will enable the researchers to quantitatively understand the vegetation changes and their trends. That information will help plan the field observations and conservation activities more effectively. Our methods facilitate cost-efficient monitoring of alpine vegetation and understanding of the progressing impacts of climate changes on alpine ecosystems.

References

[1] Intergovernmental Panel on Climate Change (IPCC). *Climate Change 2007: ThePhysical Science Basis: Contribution of Working Group I to the Fourth AssessmentReport of the IPCC.* Cambridge University Press, 2007.

[2] Jake M. Alexander, Jonas J. Lembrechts, Lohengrin A. Cavieres, Curtis Daehler, Sylvia Haider, Christoph Kueffer, Gang Liu, Keith McDougall, Ann Milbau, Aníbal Pauchard, Lisa J. Rew, and Tim Seipel. Plant invasions into mountains and alpine ecosystems: current status and future challenges. *Alpine Botany*, 126:89–103, 10 2016.

[3] Gaku Kudo, Yukihiro Amagai, Buho Hoshino, and Masami Kaneko. Invasion of dwarf bamboo into alpine snow-meadows in northern japan: pattern of expansion and impact on species diversity. *Ecology and Evolution*, 1:85–96, 9 2011.

[4] Yukihiro Amagai, Masami Kaneko, and Gaku Kudo. Habitat-specific responses of shoot growth and distribution of alpine dwarf-pine (pinus pumila) to climate variation. *Ecological Research*, 30:969–977, 11 2015.

[5] Gaku Kudo, Mitsuhiro Kimura, Tetsuya Kasagi, Yuka Kawai, and Akira S. Hirao. Habitat-specific responses of alpine plants to climatic amelioration: Comparison of fellfield to snowbed communities. *Arctic, Antarctic, and Alpine Research*, 42:438–448, 11 2010.

[6] Susana Baena, Justin Moat, Oliver Whaley, and Doreen S. Boyd. Identifying species from the air: Uavs and the very high resolution challenge for plant conservation. *PLOS ONE*, 12:e0188714, 11 2017.

[7] Andrew D. Richardson, Bobby H. Braswell, David Y. Hollinger, Julian P. Jenkins, and Scott V. Ollinger. Near-surface remote sensing of spatial and temporal variation in canopy phenology. *Ecological Applications*, 19:1417–1428, 9 2009.

[8] Dawn M. Browning, Jason W. Karl, David Morin, Andrew D. Richardson, and Craig E. Tweedie. Phenocams bridge the gap between field and satellite observations in an arid grassland ecosystem. *Remote Sensing 2017, Vol. 9, Page 1071*, 9:1071, 10 2017.

[9] Reiko Ide and Hiroyuki Oguma. A cost-effective monitoring method using digital time-lapse cameras for detecting temporal and spatial variations of snowmelt and vegetation phenology in alpine ecosystems. *Ecological Informatics*, 16:25–34, 7 2013.

[10] D. M. Woebbecke, G. E. Meyer, K. Von Bargen, and D. A. Mortensen. Color indices for weed identification under various soil, residue, and lighting conditions. *Transactions of the ASAE*, 38:259–269, 1 1995.

[11] Penelope How, Nicholas R.J. Hulton, Lynne Buie, and Douglas I. Benn. Pytrx: A python-based monoscopic terrestrial photogrammetry toolset for glaciology. *Frontiers in Earth Science*, 8:21, 2 2020.

[12] Celine Portenier, Fabia Hüsler, Stefan Härer, and Stefan Wunderle. Towards a webcam-based snow cover monitoring network: Methodology and evaluation. *Cryosphere*, 14:1409–1423, 4 2020.

[13] Pablo F Alcantarilla, Jesús Nuevo, and Adrien Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *Proceedings of the British Machine Vision Conference*, 2013.

[14] Nguyen Thanh Son, Chi Farn Chen, Cheng Ru Chen, Huynh Ngoc Duc, and Ly Yu Chang. A phenology-based classification of time-series modis data for rice crop monitoring in mekong delta, vietnam. *Remote Sensing 2014, Vol. 6, Pages 135-156*, 6:135–156, 12 2013.

[15] Jan Tigges, Tobia Lakes, and Patrick Hostert. Urban vegetation classification: Benefits of multitemporal rapideye satellite data. *Remote Sensing of Environment*, 136:66–75, 9 2013.

[16] Katharina Heupel, Daniel Spengler, and Sibylle Itzerott. A progressive crop-type classification using multitemporal remote sensing data and phenological information. *PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 86:53–69, 4 2018.

[17] Giorgos Mountrakis, Jungho Im, and Caesar Ogole. Support vector machines in remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66:247–259, 5 2011.

[18] Alex Graves, Abdel Rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 6645–6649, 10 2013.

[19] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks, 6 2014.

[20] Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. pages 338–342, 2014.

[21] Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. Joint language and translation modeling with recurrent neural networks. 10 2013.

[22] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1724–1734, 6 2014.

[23] DIno Ienco, Raffaele Gaetano, Claire Dupaquier, and Pierre Maurel. Land cover classification via multitemporal spatial data by deep recurrent neural networks. *IEEE Geoscience and Remote Sensing Letters*, 14:1685–1689, 10 2017.

[24] Atharva Sharma, Xiuwen Liu, and Xiaojun Yang. Land cover classification from multi-temporal, multi-spectral remotely sensed imagery using patch-based recurrent neural networks. *Neural Networks*, 105:346–355, 9 2018.

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 11 1997.

[26] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *32nd International Conference on Machine Learning, ICML 2015*, 1:448–456, 2 2015.

[27] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 6 2016.

[28] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020)*, 8 2020.

[29] A. Messerli and A. Grinsted. Image georectification and feature tracking toolbox: Imgraft. *Geoscientific Instrumentation, Methods and Data Systems*, 4:23–34, 2 2015.

[30] Szabolcs Dombi. Moderngl, high performance python bindings for opengl 3.3+.

[31] Juyang Weng, Paul Cohen, and Marc Herniou. I i ieee transactions camera calibration with distortion models and accuracy evaluation. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 14, 1992.

[32] Nikolaus Hansen, Sibylle D. Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary Computation*, 11:1–18, 3 2003.

[33] Zhuoqian Yang, Tingting Dan, and Yang Yang. Multi-temporal remote sensing image registration using deep convolutional features. *IEEE Access*, 6:38544–38555, 7 2018.

[34] Guorong Wu, Minjeong Kim, Qian Wang, Yaozong Gao, Shu Liao, and Dinggang Shen. Unsupervised deep feature learning for deformable registration of mr brain images. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 16:649, 2013.