Project Report: Height Estimation of Features in the "Wild"

Group 16: Simon Seo (myunggus), Eileen Li (chenyil)
16-822: Geometry-based Methods in Vision, CMU

Fall Semester: December 2023

1 Introduction

Simon and Eileen are both avid hikers. We often find ourselves in the middle of mountainous terrain without cell signal, in awe of the peaks in our surroundings. More than once we've wondered, "How tall? How far?". It would be extremely useful to approximate the height of various features we see, using nothing more than a few images from our mobile phones. In addition to satisfying our curiosities, it could also help us make more informed decisions in the unforgiving outdoors.

We wish to leverage the techniques we learned in class as applied to this personally motivated problem. Our project goal is to develop **a system for estimating the height of a standing object using geometry-based methods in computer vision** (i.e. no sensors and no learning). Our method must only use tools that are available to hikers in the wild, and in particular we focus on using stereo images from a mobile phone. However, we do make some assumptions: that the image has required metadata attached (i.e. longitude, latitude, altitude, and image direction). We use this information to fix the scale ambiguity inherent in our initial 3d scene reconstruction and are able to output height estimations within acceptable margins of error.

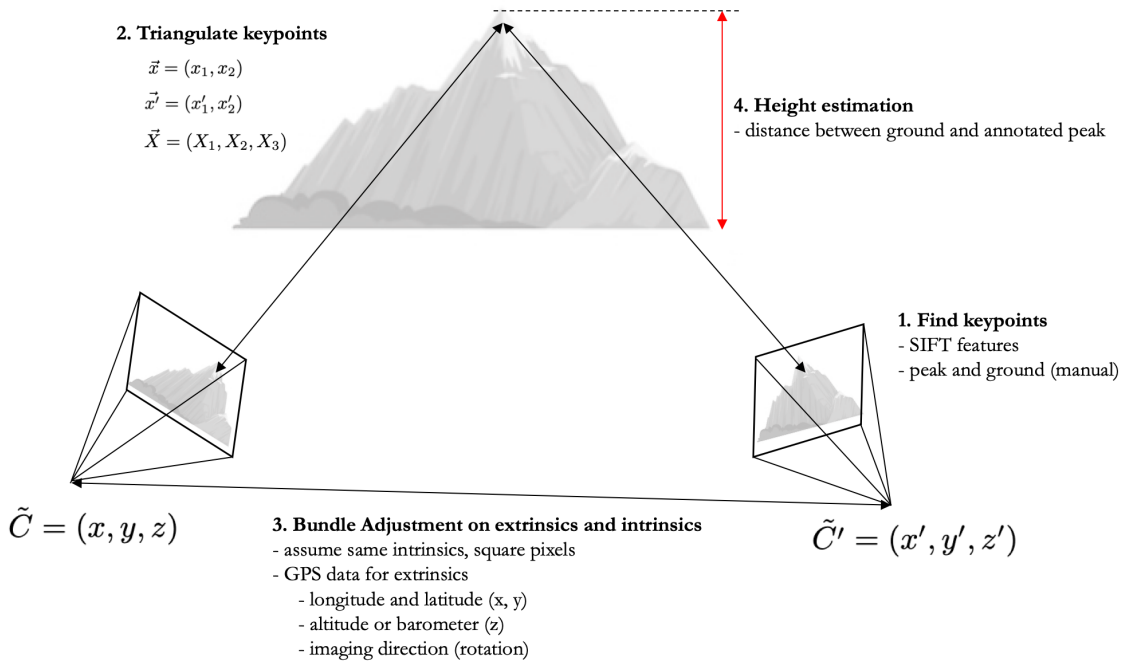

Figure 1: Problem setup. Red arrows indicate the unknown variables that we aim to estimate.

We will tackle sub-goals in the following order of increasing difficulty:

1. Height estimation of a person using a known object calibration.
2. Height estimation of a building with GPS calibration.
3. Height estimation of a mountainous feature in the "wild" with GPS calibration.

The code can be found at: https://github.com/eileenforwhat/height_estimation_ITW.

2 Related Work (mostly repeat of proposal)

The body of work on 3D reconstruction is large and diverse. Below we explore a few prior works related to our project goals: 3D reconstruction from hand-held cameras (i.e. smartphone) and height estimation using neural methods (of pedestrain and trees).

2.1 3D reconstruction with hand-held cameras

Sato et al. [2] presents a method for outdoor scene reconstruction from videos captured by a smartphone. The authors propose a method that uses multiple image sequences (i.e. videos). First, camera parameters are estimated for each image sequence by tracking annotated markers and natural features. Then at each frame, a dense depth map is computed. Finally, a 3-D model is reconstructed by combining hundreds dense depth maps in a voxel space.

While we are not going to compute a dense depth map for our scenes, we draw inspiration from techniques in this paper that deal with outdoor scenes, such as matching natural features, color adjustment among images, considering occlusions, handling regions without informative textures, etc.

2.2 Height Estimation using Neural Methods

Shen et al. [3] tackle the problem of tree height estimation using ARCore and MidasNet for depth estimation. The authors use Attention-UNet to segment the object of interest and extract axis-parallel bounding boxes in order to make the height measurements. Kim et al. [1] first estimates the ground plane as the reference plane before calculating vertical height of objects on the plane using a encoder-decoder network. To make their neural technique robust, they apply a height consistency loss on each object across reconstructions from different viewpoints.

Similarly, we may apply segmentation models and ground estimation techniques for more robust height estimation. Unlike these two papers, we will not be using out-of-the-box depth estimation tools such as ARCore or MidasNet or any neural-based learning. Instead we will rely on geometry-based methods to perform 3d reconstruction.

3 Method

In this section we detail a robust pipeline for height estimation in the wild:

1. Stereo image capture of target objects. We obtain sets of varying difficulty, from person to building to mountain.
2. Uncalibrated reconstruction. Find correspondences between the two images using robust feature detection (i.e. SIFT) and RANSAC to account for noise. Estimate fundamental matrix **F** from above correspondences and camera matrices **P1** and **P2** for the uncalibrated case. Perform triangulation of target object to obtain points in 3D, *up to a projective ambiguity*.
3. Bundle adjustment to resolve projective ambiguity using GPS and view direction, with additional camera assumptions as constraints.
4. Height estimation with known GPS distance of baseline (distance between **C1** and **C2**) to obtain correct scale.

The key to viability of our method in the outdoor context falls in the last two steps, which uses metadata attached to images captured by our mobile phones to calibrate the 3D reconstruction to the real world.

3.1 Dataset Collection

The first step is to capture stereo images. Below are some requirements that would aid in better results:

- Features: There must be sufficient overlap between the two images for correspondences to work well. This means we cannot take images from completely opposite angles, for example.
- Unobstructed views: the two images should be free from any direct obstructions that could hinder the visibility of the target object.
- Sufficient baseline: reconstruction would fail in the case of a pure rotation.

For a pair of stereo images, we can extract from the EXIF file: the **GPS location** (i.e. longitude, latitude, and altitude) and therefore the relative camera locations, as well as the **image direction** and therefore the relative camera rotations.

Note that while the EXIF metadata also contains the focal length (mm), this is not the pinhole focal length as needed by the camera intrinsics **K**, but instead the lens focal length and is not directly useful as an additional constraint in our calibration process.

3.2 Uncalibrated Reconstruction

The steps of two-view reconstruction: 1) finding correspondences, 2) estimating **F** (uncalibrated), 3) recovering camera matrices, and 4) computing 3D via triangulation, follows the procedure we've seen in lecture and in our assignments. So while we won't go into every step in detail, we highlight some practical challenges and concerns while implementing this for our project.

Obtaining accurate 2D correspondences is paramount in the accuracy of our results. We experimented with manually annotating correspondences for the 8-point algorithm compared with using SIFT with top-2 knn ratio test and found that the latter was more robust (and less manually exhaustive!). Additionally, while estimating our fundamental matrix **F**, we normalize our points via zero-center and unit variance transformation $T = \begin{bmatrix} s & 0 & -sx_0 \\ 0 & s & -sy_0 \\ 0 & 0 & 1 \end{bmatrix}$ prior to running RANSAC. This allowed us to tune down the threshold `cv2.findFundamentalMat:ransacReprojThreshold` from the default value of 3 to 0.001 and obtain better correspondences for our reconstruction. We perform triangulation of 2D to 3D using only the inliers from the previous step.

The key thing to note is that our solution for camera matrices P_1 and P_2 has an ambiguity up to a homography H such that (P_1H, P_2H) is also a solution, causing our results to differ from the ground truth by orders of magnitude.

3.3 Bundle Adjustment with Constraints

The way to resolve the projective ambiguity of uncalibrated reconstruction is through bundle adjustment, where we iteratively refine both the 3D points (X_1, X_2, X_3) and the camera intrinsics K_1 and K_2 by minimizing the reprojection error with additional constraints from GPS and camera assumptions (i.e. square pixels and same K).

More formally, we jointly perform bundle adjustment and auto calibration to refine the parameters:

$$K = \begin{bmatrix} f_x & s & t_x \\ 0 & f_y & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

,

$$\mathbf{X}^{(i)} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$$

by solving the following minimization problem:

$$\begin{aligned} &\min_{X,K_1,K_2} \sum_i \left[|\mathbf{x}_1^{(i)} - \mathbf{P}_1\mathbf{X}^{(i)}|^2 + |\mathbf{x}_2^{(i)} - \mathbf{P}_2\mathbf{X}^{(i)}|^2 + |f_x - f_y|^2 + |\mathbf{K}_1 - \mathbf{K}_2|^2 \right] \\ &\text{where } \mathbf{P}_1 = \mathbf{K}_1[\mathbf{R}|\mathbf{t}], \mathbf{P}_2 = \mathbf{K}_2[\mathbf{R}|\mathbf{t}], \text{ and } s = 0 \end{aligned} \tag{1}$$

3.4 Height Estimation

The final step to our pipeline is to use the calibrated camera matrices (P_1', P_2') from bundle adjustment, as well as the true camera baseline as provided by GPS, to estimate the height of a target object. We find the scaling factor, s, from comparing the distance between computed camera centers by solving $P'C = 0$ and the distance between true camera centers using the Haversine formula (assuming flat Earth):

$$
\begin{aligned}
&R = 6371000 \quad \text{(radius of Earth in meters)} \\
&a = \sin^2\left(\frac{\Delta\text{lat}}{2}\right) + \cos(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \sin^2\left(\frac{\Delta\text{lon}}{2}\right) \\
&c = 2 \cdot \arctan 2\left(\sqrt{a}, \sqrt{1-a}\right) \\
&\text{gps\_baseline} = R \cdot c \\
&s = \frac{\text{gps\_baseline}}{\text{computed\_baseline}}
\end{aligned}
$$

The final height is the scaled 3D length between peak points (X_1, X_2, X_3) and base points (X_1', X_2', X_3'):

$$\text{height} = norm(s \cdot (X_1, X_2, X_3) - s \cdot (X_1', X_2', X_3'))$$

In our current pipeline, the peak and base points are obtained from manual 2D image annotation projected to 3D. This annotation requirement can be removed via segmentation of target object (discussed more in future work).

3.4.1 Ground Plane Estimation

We also implement ground plane estimation by annotating three points on the ground plane and computing the normal vector to this plane using cross product of vector pairs. We adjust our height estimation pipeline to calculate plane-point distance. This improves the robustness as the base of the target object cannot always reliably seen in the wild.

4 Experiment Results

The results of the three sub-goals are shown. For each target object, we calculate the error between our estimation and the ground truth. The error increases with each step in difficulty, though the highest is still within an impressive 13%.

Target	Method	Prior	Prior Length	Estimated Height	GT Height	Error
Person (Simon Seo)	Two-view reconstruction without rectification	Cereal box	0.31m	1.82m	1.75m	4%
Building (Cathedral of Learning)	Ground plane estimation, Two-view reconstruction, bundle adjustment	GPS baseline	35m	146m	163m	10%
Mountain (Banner Peak)	Two-view reconstruction, bundle adjustment	GPS baseline	266m	295m	261m	13%

Table 1: Height estimation results.

4.1 Height Estimation: Person

To simplify the problem, we first estimated the height of a person with uncalibrated reconstruction, but using a prior object of known length to disambiguate the scale. The prior object and target object must additionally be on the same plane to remove need for rectification and provide accurate results. We used this step to verify our feature extraction and reconstruction modules.

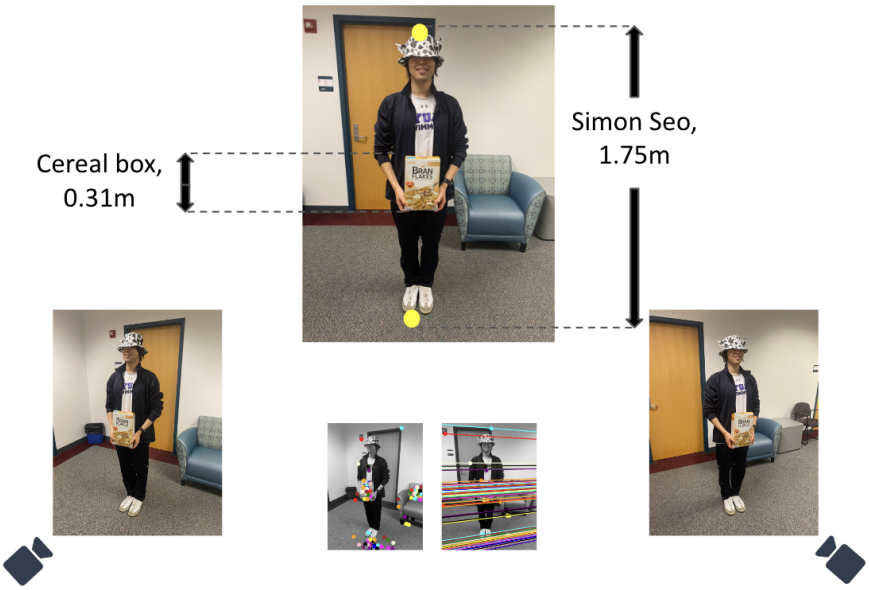

Figure 2: Stereo setup for height estimation with prior object.

4.2 Height Estimation: Building

As an intermediate step before estimating height of a mountain, we test our method on a local building which provides similar types of conditions as that of a mountain. In this step we utilize a GPS-informed baseline as the metric prior, instead of an object of known length. Since the known prior and the target are on different planes, we must perform bundle adjustment jointly with auto calibration using non-linear constraints on the intrinsic matrix $\mathbf{K}$ (see Eq 1).

An additional challenge was data collection. The urban setting posed a lot of obstructions at a distance and strong projective distortion up close. We found an open field where we had a clear view of the base and the peak of the building. Note that this is a limitation of our method, which requires a view of the ground in order to estimate its location, either through manual point annotation or plane detection.

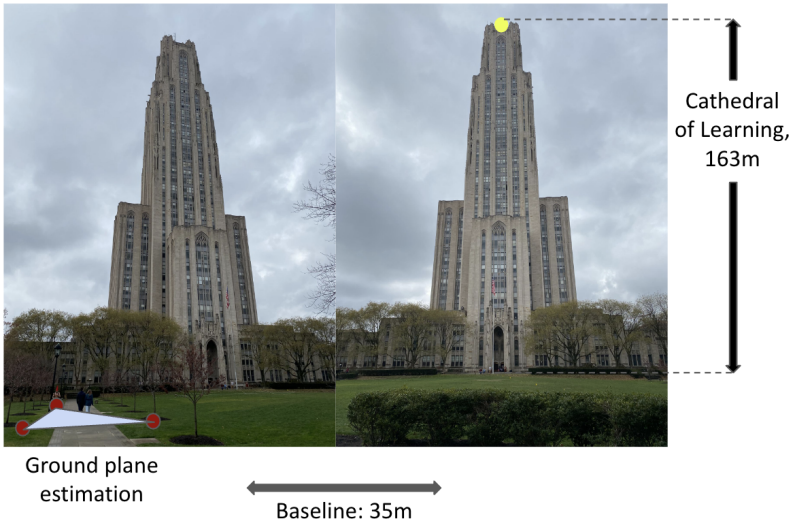

Figure 3: Building height estimation with GPS distance as baseline and ground plane estimation.

4.3 Height Estimation: Mountain

As a final challenge, we apply our pipeline to mountainous peaks. However, due to the lack of mountainous terrain around Pittsburgh and the limitation that online images lack many of the required EXIF data, we look to past image captures instead. Luckily, we were able to find many suitable stereo images in our photos history, one of which is shown below.

One concern we had was the quality of correspondence matches in this new outdoor domain. The entire pipeline worked surprisingly well, even with a much larger baseline than previously.

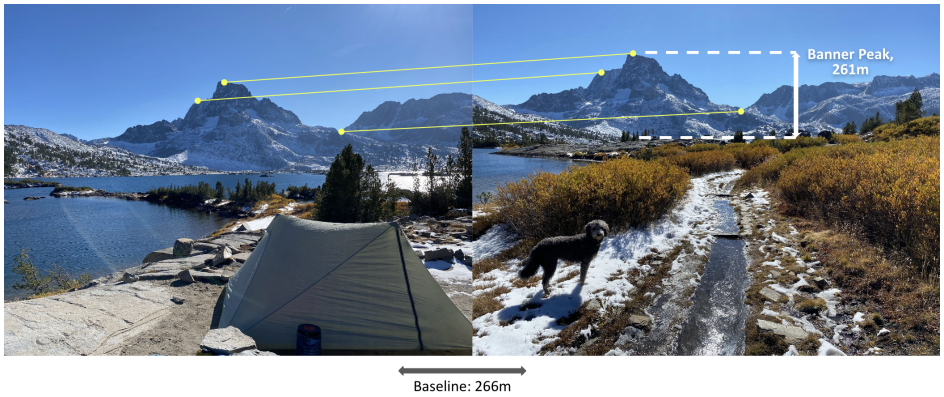

Figure 4: Mountain height estimation with GPS distance as baseline.

5 Conclusion and Future Work

We have described a reliable way to estimate the height of natural features in the wild. We do so using only two captured images and GPS information from our mobile phones, a reflection of the limited outdoor context we are targeting. In the current version, we rely on manual annotations (of peak/base and/or ground) to capture the distance we wish to estimate. A future direction would be to replace all dependencies on manual annotation, as this is also our main source of estimation error. A segmentation model can be used to capture the peak location, and the ground plane can be approximated from stereo depth maps. A more ambitious extension to our algorithm would be to use short real-time video rather than stereo images, which has potential to improve both the accuracy and user experience. With the techniques outlined in this report, hikers everywhere can rejoice in always being able to answer the question “how high are you?”.

References

[1] In Su Kim, Hyeongbok Kim, Seungwon Lee, and Soon Ki Jung. Heightnet: Monocular object height estimation. *Electronics*, 12(2), 2023.

[2] T. Sato, M. Kanbara, and N. Yokoya. Outdoor scene reconstruction from multiple image sequences captured by a hand-held video camera. In *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI2003.*, pages 113–118, 2003.

[3] Yulin Shen, Ruwei Huang, Bei Hua, Yuanguan Pan, Yong Mei, and Minghao Dong. Automatic tree height measurement based on three-dimensional reconstruction using smartphone. *Sensors*, 23(16), 2023.