Lidar assisted Depth Estimation for Thermal Cameras

Bachelorarbeit

vorgelegt von

Julian Jandeleit

an der

Universität Konstanz

Mathematisch-Naturwissenschaftliche Sektion

Fachbereich Informatik und Informationswissenschaft

1. Gutachter: Prof. Dr. Bastian Goldlücke

2. Gutachter: Prof. Dr. Oliver Deussen

Konstanz, 2022

Abstract

Depth- and pose estimation are classical problems in computer vision. Estimating the depth of thermal cameras can be achieved by estimating the camera poses in an independently captured lidar scan.
In underground environments and animal behavior, thermal cameras are often used instead of classical RGB cameras. But existing methods for RGB pose estimation do not necessarily need to be effective for thermal cameras, because their working mode and data are different. This is the case for subterranean cave environments. Thermal cameras can be used to capture animal behavior in dark caves. This is what this paper is focused on. This work provides a systematic approach to the alignment of lidar and thermal image information. We use the Ushichka dataset as an exemplary study case. Classical computer vision techniques like Direct Linear Triangulation and Space Carving are are applied to the dataset. We find that the classical methods do not work here as expected and provide a semi-automatic framework, called DMCP, to solve the registration. Finally, different approaches are compared in order to point out in which way a fully automatic registration might be possible.

Contents

1 Introduction

For modeling bat behavior in their indigenous environment, 3D information of their native cave is needed. Cameras used to observe the animals do not capture depth directly.
Depth estimation is the task of extracting 3D Information from one or more two dimensional images of the scene. The estimated depth assigns each pixel in the image the distance of the world point whose color was captured in the respective pixel. The estimated depth is associated with the shape observed by the cameras and thus is inherently a problem of 3D reconstruction. However, multiple points can project to the same point in the image. This shows that 3D reconstruction from images alone is generally ill-posed.
A lidar sensor sends laser rays into the world and measures the time the reflected signal needs back to the sensor. The time information is then used to calculate the distances to the points where the individual points were reflected [9]. Figure 1 illustrates the basic principle behind lidar sensors. Lidar is effective at accurately measuring points in a 3D Structure. However, despite recent cost improvements, lidar is a more costly technology than cameras [26]. Also, lidar can generates comparatively large data sets. Additionally, due to the nature of the sensors, accurately scanning a complete scene, can may take some time to finish [9].
Thermal cameras, on the other hand, observe heat emitted by the objects observed. In particular, the cameras capture infrared radiation. Infrared radiation is one of the three ways heat is transmitted and is sent out by all objects to some degree. Therefor thermal cameras are optimal in low light conditions. They can observe objects having a heat signature different than the surrounding temperature. This allows thermal cameras for example to observe animals in complete darkness.
When using lidar, we can capture structure of the surrounding area which stays the same over time. And we can capture the real time behavior of moving objects using thermal cameras. The information can then be integrated by finding the registering transformation that aligns the cameras and the lidar scan.

In this paper we focus on data from the Ushichka Dataset [25]. It consists of a lidar scan of the Orlova Chuka cave in Bulgaria. The cave if known as a home for bats. Three thermal cameras observe a room in the cave, primarily to capture the flight behavior of the animals. On the images, the bats stand out as a bright point in the darker, comparatively homogenous background. The relative position of the cameras are known, as well as their intrinsic calibration.
First step is to find the position of the cameras in the lidar scan. Based on this the flight paths of the bats can be reconstructed. Then the estimated position can be used to view the paths relative to the cave walls, which might spark biological insight about bat behavior [25]. The position in the lidar scan can be interpreted as an affine transformation that aligns the cameras to the scan. In particular, we want to find structure in the images of the thermal cameras. The information found in the structure should be employed to calculate the registering transformation for each camera.
To reach this target, we will:

- Outline different approaches and highlight previous work which specializes on pose estimation and 3D reconstruction, especially for thermal images.
- Introduce the semi-automatic DMCP algorithm for direct pose estimation.
- Experimentally evaluate classical approaches for computer vision on the Ushichka dataset.

The focus is classical methods in computer vision, meaning that machine learning techniques are not considered.

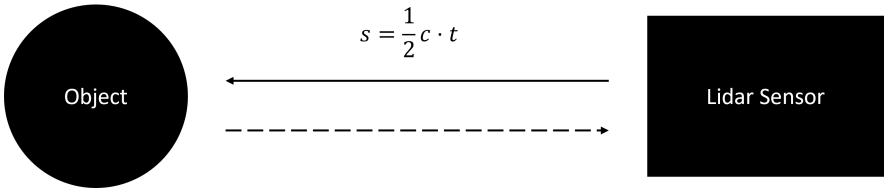


Figure 1: Basic principle behind lidar sensors. The lidar sensor sends a laser pulse into the scene. It travels until it gets reflected by an object. The sensor recognizes the reflected pulse and stops the time t between exit and entrance. With the constant velocity of light c, the recorded time directly relates to the measured distance s to the object by $2 \cdot s = c \cdot t$.

2 Background to Computer Vision

Luminance Model For this thesis, we consider the scene to be fully lambertian. The term lambertian refers to an ideal diffusely reflecting surface. The brightness of a lambertian surface is determined entirely by the intrinsic value of the surface, independent of the position and orientation of the observer. The light is emitted equally in every direction [13]. In this paper, we use the terms color, radiance, pixel value and intensity synonymously. They all refer to the value, the camera recorded at a certain pixel. This value is assumed to have been emitted by the scene point. The point itself has the same value as its intrinsic color.
For thermal cameras, in fact we capture infrared radiance and thus heat instead of visible light.

A Note on Coordinate Systems In this paper, we distinguish between several coordinate systems.
First, there is the *world space*. It contains all objects in the scene. It is assumed to be the space in which position and distances are measured in reality.
Second, there is the *lidar space*. It is the coordinate system which is native to points in the lidar scan. For simplicity, we assume lidar and world space to be equal.
Third, there are (thermal) cameras distributed throughout the scene. They are assumed to be calibrated to each other. This means that their (relative) position and orientation is known in some coordinate system. We call it *calibration space*. Note that, if calibration space is equal to lidar space, the problem is solved trivially.
Finally, each camera has its respective coordinate system which is called *camera space*. The origin of the space is the cameras center. The camera space describes how the camera views the world. Similarly, there is the *pixel space* which is the plane where the image forms.
Figure 2 shows the relationship between the individual spaces.

Homogeneous Coordinates Points belonging to either of above spaces may be given in euclidean coordinates $\mathbb{R}^n$ or in homogeneous coordinates $\mathbb{P}^n$. $\mathbb{P}^n$ is called the n-dimensional projective space [1].
$v_{\mathbb{R}} = [x, y, z]^T \in \mathbb{R}^3$ relates to $v_{\mathbb{P}} \in \mathbb{P}^3$ by $v_{\mathbb{P}} = [x, y, z, 1] \in \mathbb{P}^3$. Additionally, there exists the equivalence class $v_{\mathbb{P}} = k \cdot v_{\mathbb{P}} = [K \cdot x, k \cdot y, k \cdot z, k]$ for $k \in \mathbb{N}$.
When its clear from context, we will not distinguish homogeneous and inhomogeneous coordinates for readability. They can be converted into each other using the relationship described above. Other number of dimensions work equivalently.

Pinhole Camera Model For computer vision, a model is needed how to convert points in world space to pixel space. Here, we assume the Pinhole Camera Model. A detailed explanation of the model can be found in *Multiple View Geometry in Computer Vision* [7]. In the following, the relevant terms are recalled shortly.

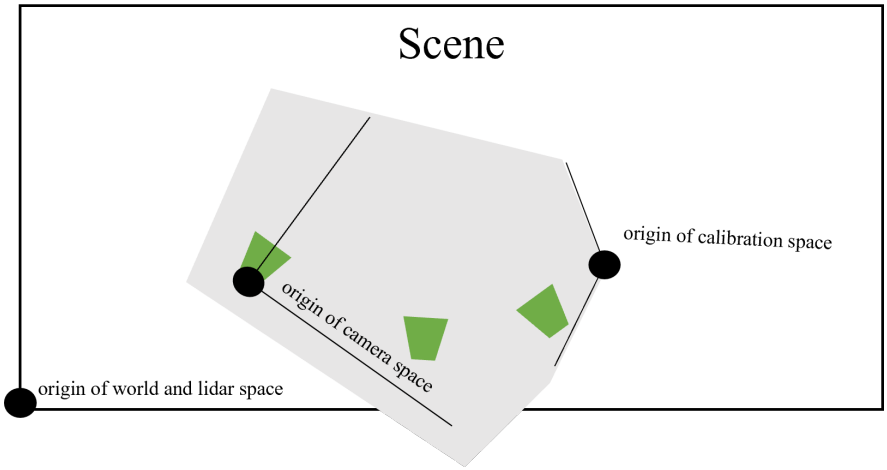


Figure 2: Relationship of different coordinate systems. The world and lidar space is seen as the all-encompassing *true* coordinate system. The cameras, shown in green, are part of the scene and have coordinates in world space. Finding these coordinates is the goal of the registration. The cameras are also enclosed in the calibration space. This is the coordinate system where the extrinsic camera properties, for example positions, are known. As with the cameras, the transformation from world to calibration space and back is not given. The axes of the calibration space are drawn in-perpendicular to indicate that there can be a skew factor between the coordinate systems. Finally, each camera has its own camera space. The space is indicated for the left camera exemplary. One axis is drawn aligned with with the viewing direction of the camera to indicate that it observes the world through this space. The pixel space is left out in this figure for simplicity.

We call $\bar{E}_c = [\bar{R}_c \mid C_c] \in \mathbb{R}^{3x4}$ the *camera pose matrix* of camera c, where $\bar{R}_c \in \mathbb{R}^{3\times3}$ describes the rotation of c and $C_c \in \mathbb{R}^{3\times1}$ describes the position of c in world space. The position is sometimes referred to as translation. $\bar{E}_c$ describes the position and rotation of camera c. The matrix transforms points from camera space to world space. The inverse matrix $E_c := \bar{E}_c^{-1}$, called *extrinsic matrix* of camera c, describes the transformation from camera to world space. The matrix $E_c = [R_c \mid T_c]$ also has two components $R_c = \bar{R}_c^T$ and $T_c = -\bar{R}_c \cdot C_c$. However, they don't describe the pose of the camera c but the world space as seen from camera space. Let $K_c \in R^{3x3}$ be the *intrinsic matrix* of camera c. It describes the transformation from camera to pixel coordinates.
In total, the *projection matrix* $P := K_c \cdot E_c = K_c \cdot [R_c \mid T_c]$ transforms points from world to pixel space. Note, that we assume homogeneous coordinates. So, if $v \in \mathbb{P}^3$ is a point in world space, it projects to $p \in \mathbb{P}^2$ in pixel coordinates with $p = P \cdot v$. When lifting points to and from projective space, P represents how points are projected in a camera. This is exactly what we expect for a camera model. The transformation is called projection.

3 Systematic of Lidar-Thermal Alignment

In this section, several approaches on how to combine lidar and thermal information are explored. They have in common that every method aims to extract and match information from both sources in order to find a transformation that aligns lidar and calibration space. The alignment is expected to minimize the distance of the projection of a point in the lidar scan to its observed location in each thermal image [4]. We call this the *alignment condition*.

3.1 Direct Alignment

The alignment conditionshows that there is a straight forward way to solve the pose estimation, if there are known correspondences. They can be achieved using calibration patterns that can be recognized in lidar data as well as in thermal data. Truong et al. [17] register point clouds generated from thermal and rgb images with a calibration chessboard pattern that is recognizable by both thermal and rgb cameras. Similarly, Kim and Park [21] use a chessboard for camera to lidar calibration.
However, calibration patterns are not always feasible or just not present such as in the Ushichka dataset [25]. Kang and Doh [24] describe a method to calibrate a camera to a lidar scan by aligning edges in the point cloud with edges in the image. Their method does not require a calibration target. However, they focus on a 360° camera which produces a spherical panoramic image.
To the best of my knowledge, there is no published approach that focuses on direct thermal camera and lidar alignment for subterranean environments without using a calibration object.

3.2 Shape Based Alignment

Instead of directly using the alignment condition, an indirect approach can also be used to solve it. The lidar scan defines a 3D shape in world space. However, images only capture a 2D projection of the true shape. If it is possible to reconstruct the 3D shape from the images, the shapes can be compared and aligned directly for example using the ICP-Algorithm [3]. In the following, we will discuss two approaches for 3D reconstruction.

Reconstruction by Feature Correspondence This approach measures interesting points in the image using feature descriptors. In general, descriptors hold some values in order to be matched against descriptors in other images to form feature correspondences. When you have correspondences of the same point in multiple images, the 3D point can be computed by solving the pinhole projection model for the point in world space. This method of triangulation is called direct linear transformation and is laid out in detail by Hartley and Zisserman [7]. Direct linear triangulation leads to a unique 3D point that minimizes the alignment condition. The camera pose estimation will then be solved by the transformation that registers the lidar shape with the shape computed from the correspondences.
Hajebi and Zelek [10] apply reconstruct feature correspondences from images of an office scene. Specifically, they use classical feature detectors and descriptors like SIFT[8]. They also use a measure called phase congruency to detect feature points. It was shown by Kovesi [6] that phase congruency is able to detect images and corners in images. Hajebi and Zelek [10] suggest that phase congruency detectors are superior to the SIFT detector for thermal images. There are several approaches that immerse themselves into the topic of phase congruency descriptors. HOP [15] and LGHD [14] use information at different orientations created during phase congruency computation. HOMPC [19] merges phase and magnitude information. All approaches are united by the use of histograms for feature representation.
The above approach can be described as local methods. At the opposite, there are the so called global methods. They do not look at pixel correspondences individually but optimize the matching of all image pixels simultaneously. This is done by building an energy function that often encodes additional constraints such as smoothness. It results in a depth map which assigns each pixel the distance of the viewed structure. The world points represented by the depth map then satisfy the alignment conditionmodel for all images. Hirschmuller [11] is a primary representation for those methods.
Nevertheless, global methods also look for some kind of correspondence in images. It can only be successful if sufficient structure is present in the images.

Reconstruction by Photoconsistent Voxels Instead of directly looking for correspondences and then triangulating the point in world space, you can also consider world points first and estimate weather the images support a specific point belonging to the surface of a structure captured by the cameras. The points are called voxels. An early work based on voxels, is "A Theory of Shape by Space Carving'' [5]. The authors show that pictures of a scene define constraints which are satisfied by all 3D scenes which project to the photographs. A central term is the concept of photo-consistency: Points that are visible at a camera, have a color equal to the points radiance at its projection. They extend the definition to complete shapes and call the union of all photo-consistent shapes the photo hull. The photo hull can then be computed by repeatedly evaluating every surface voxel for photo-consistency and removing ("carving") inconsistent ones.
To calculate the photo hull more efficiently, they introduce a sweep plane that moves across the voxel volume. Pixels which correspond to a consistent voxel get marked so that only unmarked get considered as photo-consistent. This solves the ordering problem for cameras behind the plane. Thus, only surface voxels intersecting with the plane need to be considered. They show that sweeping a plane along every axis and axis-direction then computes the photo hull efficiently. This is called the Multi-Sweep Space Carving Algorithm.

4 Depth-Map-Correspondence Algorithm

As the Ushichka dataset does not have a calibration target, we will lay out a custom way for direct calibration that only uses the lidar scan and calibrated thermal image information. The algorithm can than be applied as ground truth for other approaches to compare compare with.

In the following, we will make use of the alignment conditiondescribed in to create a semi-automatic method for direct alignment. The algorithm will be referred to as DMCP in short for *depth-map-correspondences*. As input, it requires a depth map from an arbitrary but known camera in the lidar scene, a (thermal) camera in its calibrated coordinate system and correspondences between the camera image and depth map in pixel coordinates. Note, that the depth map can be computed by backprojecting the mesh into a virtual camera, that observes the scene.
From this input, an affine transformation A that registers points in calibration space with points in lidar space will be computed. Figure 3 shows the general flow of information as an outline of the algorithm. The algorithm can roughly be divided into two parts:

1. Pose calibration for one camera
2. Computing the registering transformation

Naming Conventions The following conventions are used regarding the meaning of symbols:
Let $P = K \cdot E$ be the projection matrix in the sense of the pinhole projection model. Then K is the intrinsic camera matrix and E is the extrinsic camera matrix, which transforms world to camera coordinates. We will call $C = E^{-1} = [R\ T]$ the extrinsic matrix which describes the transformation from camera to world coordinates. R describes the rotation and T the position of the camera in world space.
We will distinguish different cameras using subscript and coordinate systems using superscript.

4.1 Pose calibration

Let P_{th}^{calib} be the projection matrix of a selected *thermal* camera, given in *calibration* coordinates. Let P_{dm}^{lidar} be the projection matrix for the camera that observed the *depth map*, given in *lidar* coordinates. Let K_{th} and K_{dm} be their respective intrinsic camera matrices. Let CPS_{th} be the ordered set of corresponding points in the thermal image and CPS_{dm} the respective points in the depth map.

Let $I_{dm}(x, y)$ denote the depth value in the depth map at position (x, y). We can now compute the 3D locations of the corresponding points CPS_{lidar} in the lidar space using the intrinsic camera matrix K_{dm} and the cameras extrinsic matrix C_{dm}.

$$CPS_{lidar} \coloneqq \{\, C_{dm} \cdot I_{dm}(x, y) \cdot K_{dm}^{-1} \cdot [x, y, 1]^T \mid [x, y, 1]^T \in CPS_{dm} \,\} \tag{1}$$

The next step is to estimate the pose of the thermal camera in the lidar space. The corresponding point sets CPS_{th} and CPS_{lidar} represent the problem of camera calibration, which can be solved using direct linear transformation (DLT) .

$$P_{th}^{lidar} \coloneqq DLT(CPS_{th}, CPS_{lidar}) \tag{2}$$

P_{th}^{lidar} represents the estimated pose for the selected thermal, camera th, in the form of a pinhole projection matrix. Absolute pose and orientation can be derived using the relationships stated above in paragraph *Naming Conventions*.

4.2 Registering Transformation

To transform any points in calibration space to lidar space, a transformation that registers those points needs to be computed.
Both coordinate systems may have a different scale, which needs to be taken account for. It can be extracted from comparing the estimated projection matrix P_{th}^{lidar} with the calibrated P_{th}^{calib}, more specifically their respective extrinsic camera matrices, which have the form $E = [r\ t]$. The matrix r is a rotation matrix and the norm of each column vector describes the scale in the respective axis. Let sv_{calib} and sv_{lidar} be the vector that describes the respective scales for each matrix, then

$$scale \coloneqq \frac{\|sv_{calib}\|}{\|sv_{lidar}\|} \tag{3}$$

describes the relative scale between both coordinate systems. This factor takes account for the different scales. Now we have enough information to transform CPS_{lidar} first into thermal camera space and from there into calibrated space.

$$CPS_{calib} \coloneqq \{\, C_{th}^{calib} \cdot scale \cdot E_{th}^{lidar} \cdot [x, y, 1]^T \mid [x, y, 1]^T \in CPS_{lidar} \,\} \tag{4}$$

This works, because P_{th}^{lidar} and P_{th}^{calib} describe the same camera, only in different frames of reference. We now have corresponding 3D points in lidar and calibration space. This allows us to calculate the registering transformation A by solving what is known as the *absolute orientation problem* using Horns method [2].

$$A \coloneqq [R\ T] = solveAbsoluteOrientation(CPS_{calib}, CPS_{lidar}) \tag{5}$$

It computes R, T such that $CPS_{lidar} = R \cdot CPS_{calib} + T$, as an affine matrix $CPS_{lidar} = [R\ T] \cdot CPS_{calib} = A \cdot CPS_{calib}$. The projection matrices P_i^{calib} of each camera i in the calibration space can then be transformed with:

$$P_i^{lidar} \coloneqq P_i^{calib} \cdot A^{-1} \tag{6}$$

This solves the pose estimation problem for all cameras in calibration space.

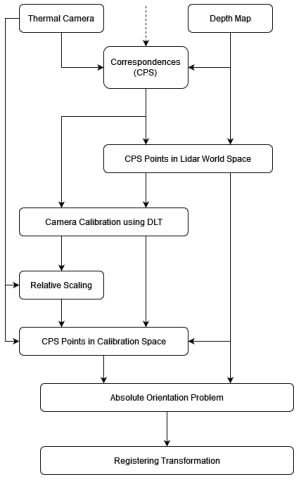


Figure 3: DMCP Information Flow

4.3 Further Considerations

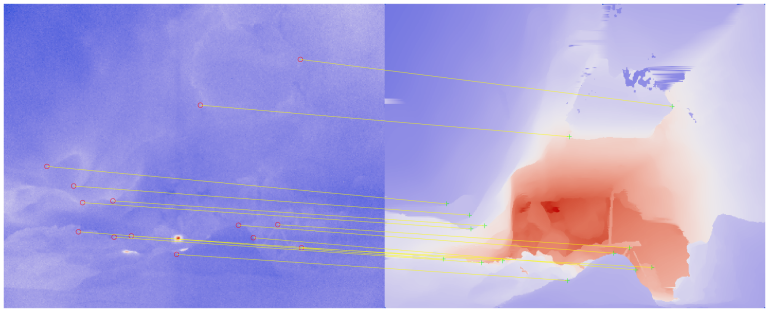
Suppose there is a method to reliably match depth map and thermal correspondences. Then, no human is necessary to apply the DCMP algorithm, apart from defining candidate camera poses for the depth map or an algorithm to to find such pose.

5 Experimental Evaluation

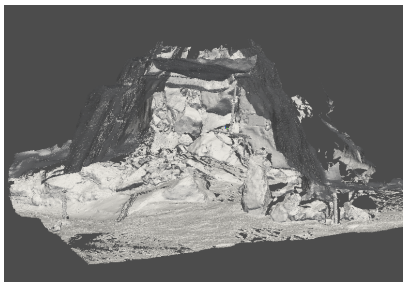
In this section, we will describe the experiments executed on the Ushichka dataset. The data contains a lidar scan of the *Orlova Chuka* cave in Bulgaria. The scan is represented in form of a mesh. Figure 4b shows the relevant part. Additionally, there are three thermal cameras. The cameras are calibrated externally and have coordinates in *calibration space*. The thermal image recordings are represented in a custom .TMC format. They are converted into a .csv file for each image using the software *ThermoViewer* [29]. They can be read in into most programming languages as a 2D array for later processing. The values in the images range from about 7000 to 7400.

Feature based reconstruction is the most favoured classical reconstruction method especially using the SIFT descriptor and detector [18]. Considered are usually RGB cameras placed in everyday situations. Due to the popularity of SIFT, the descriptor is used to try out the feature correspondence based approach. Additionally LGHD[14] features are considered to use the results from Hajebi and Zelek [10], who recommend phase congruency feature detectors for thermal images.
Finally, we apply a custom implementation of space carving and a derived approach that takes into account phase congruency.

Image Preprocessing The walls and structures all have a similar temperature which, in addition to general infrared imaging characteristics [10], result in comparatively low gradients at structure boundaries. This indicates a low signal-signal-to
In this case, we can see in Figure 5b that there also is a lot of fixed pattern noise (FPN) present. Wang et al. proposes a method for removing this kind of noise from infrared images [23]. Here, we will employ the straight forward approach to extract the FPN by selecting all horizontal and vertical components of the Fourier-Transformation of the representative image and then subtracting it from the original. Finally, representative images for each thermal

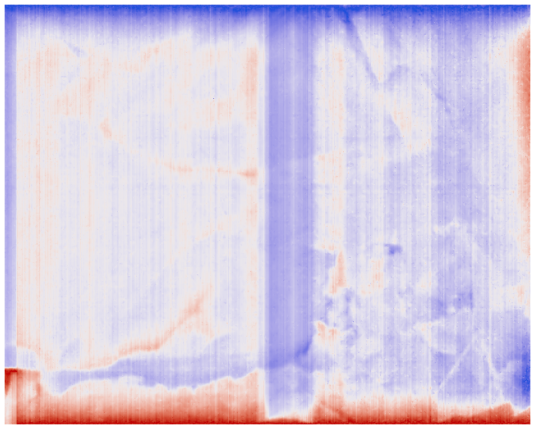


(a) Annotated correspondences between thermal image (left) and depth map (right)

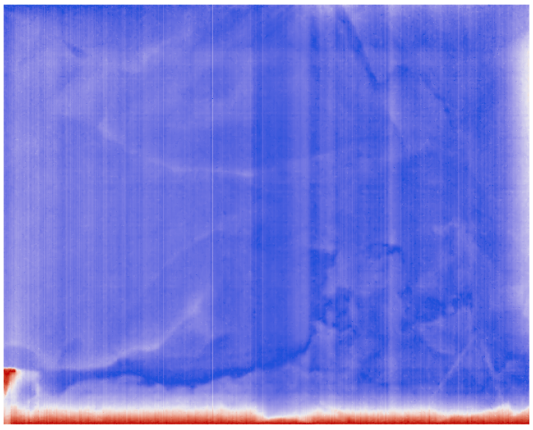


(b) Primary section of lidar mesh

Figure 4: DMCP Experiment



(a) Thermal image scaled to $[0, 1]$



(b) Thermal image without FPN, scaled to $[0, 1]$

Figure 5: Fixed-Pattern-Noise for representative image of camera 1

cameras are obtained by taking the median at each pixel along the time axis. The representative images are used as input for the experiments. The videos chosen to generate the representative images have a length of 100 frames.
Figure 5 shows the result of the FPN removal on the representative image for camera 1.

5.1 DMCP Alignment

The programming languages MATLAB [27] was used to implement the DMCP algorithm and to run the experiment. The depth-map that is required as input was computed by backprojecting the mesh to a virtual with extrinsics similar to camera 2 and the same intrinsics. The depth-map was generate using python and the PyVista library [22]. PyVista implements depth-depth map generation for OpenGL-like camera models. The correspondences from thermal camera to depth map were annotated manually and are shown in Figure 4a.

The DMCP algorithm was executed on the data specified above. As result the estimated camera positions shown in Figure 6 have been obtained. The estimated transform as well as the resulting pinhole projection matrices in lidar space can be found in the appendix at part A. There it also is demonstrated how to extract the camera position from the projection matrices. The pose estimation in lidar space can now be used to backproject the mesh into a depth map as described above. The annotated correspondences can now be used to obtain:

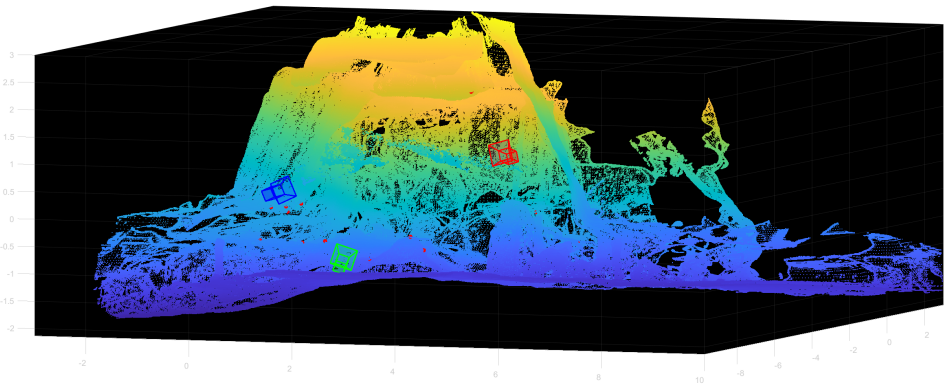


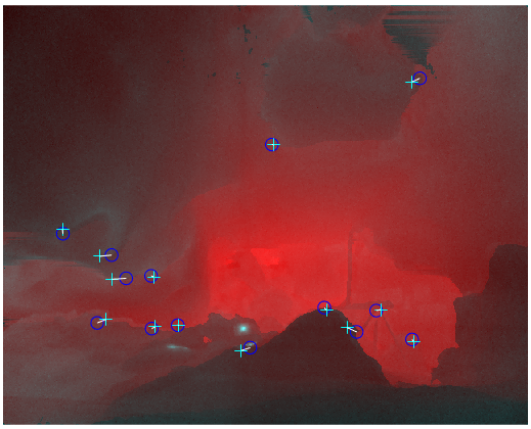
Figure 6: Estimated camera positions, transformed using transformation obtained from DMCP. Camera 1 is shown in red, camera 2 in green and camera 3 in blue. Also shows visible annotated correspondences in 3D as small red points.

1. The point in lidar space that was annotated using the input depth map.
2. The estimated point that was annotated from the depth map generated from the estimated projection matrices in lidar space.

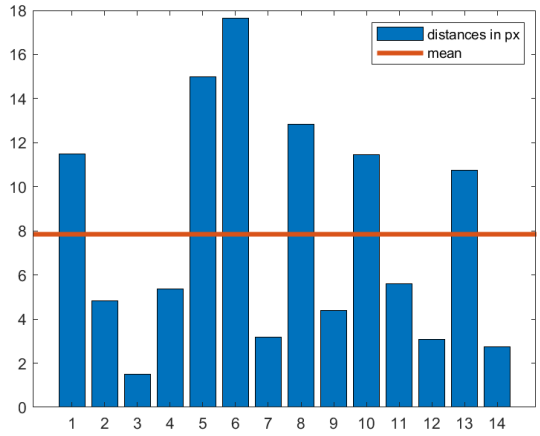
The registration error can be computed by comparing the annotated point with the estimated point for each correspondence as described with the alignment condition. On a perfectly registered camera, points on the image that correspond to the same 3D point, should lie on the same pixel coordinate. The distance in pixels between annotated points in the thermal and estimated depth image are computed. The differences are illustrated in Figure 7a. The mean distance is 7.9 pixels. The barchart of all distances in Figure 7b.
A scaling factor of about 1.65 between calibration and lidar space was estimated during the algorithm.

5.2 Reconstruction by Feature Correspondence

Tareen and Saleem [20] recommend SIFT [8] to be the descriptor that in general, for images taken with a normal camera, is the most accurate algorithm for feature correspondence based reconstruction.
Phase congruency is a measure which provides information invariant to contrast and can be used to detect edges and corners [6]. Hajebi and Zelek [10] suggests phase congruency to be used on thermal image data. The LGHD descriptor [14] is one of many proposed descriptors that describe local features using phase congruency.
In the following, two experiments are done that aim to reconstruct the scene by feature correspondence using SIFT and LGHD descriptors respectively.



(a) Overlayed thermal image (camera 2) and its estimated depth map. Points in thermal image are marked as cyan crosses, points in the corresponding depth map are marked as blue circles.



(b) Distances between points in thermal image and estimated depth map.

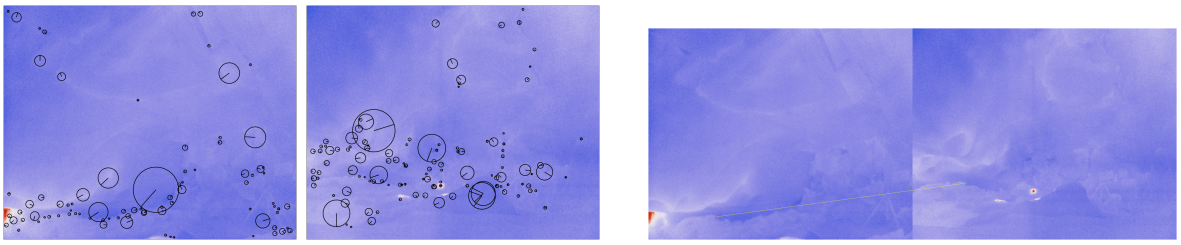
Figure 7: Evaluation DMPC

SIFT SIFT is used to detect features in the thermal images. The features are matched using Lowes method to reject ambiguous matches.
MATLAB and the VLFeat libaray [12] are used for the actual computation. The parameter *PeakThresh* is set to 0.5. Pixels are considered to be fundamental inliers if the matched point is closer that 5 pixels to the epipolar line. The epipolar line represents all points where a correspondence can be found due to the geometry of two cameras. It is calculated using the fundamental matrix and the calibrated cameras (see *Multiple View Geometry in Computer Vision* [7]).
Figures 8a and 8b show the points that are detected using SIFT. Figure 8c shows the matching inliers, that satisfy the epipolar constraint described above.
It can be seen that SIFT mostly detects points near structure boundaries. However, there are also many points distributed between boundaries. There is only one match that satisfies the epipolar constraint. When visually interpreting the match, it can be found that the matched point indeed does correspond.

LGHD The LGHD descriptors will be applied to the phase congruency feature points. The descriptors are matched using cosine similarity, as suggested by Hajebi and Zelek [10].
Phase congruency is computed using the library from Peter Kovesi [28]. The the parameters $sigma = 1.5$ and $n\_pts = 500$ are used. The LGHD descriptor is computed using the implementation by the authors. Similar to above, pixels are considered to be fundamental inliers if the matched point is closer that 5 pixels to the epipolar line.
Figures 9a and 9b show the points that are detected using LGHD. Figure 9c shows the matching inliers, that satisfy the epipolar constraint.
The detected points lie on most structure boundaries that can be identified visually by humans. There are several matches that satisfy the epipolar constraint. Nevertheless, a few of those matches can be identified as false positives, as they do not describe the same 3D point.

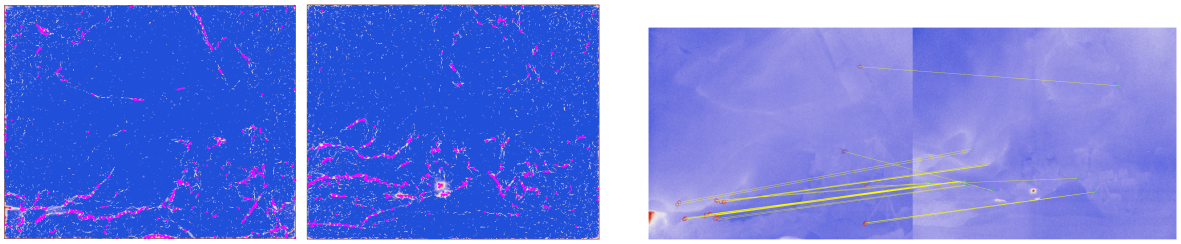
5.3 Reconstruction by Photoconsistent Voxels

Space carving is used to experiment with reconstruction by photoconsistent voxels. To build on the results of the LGHD descriptor described in section 5.2, a second experiment is performed. The second experiment represents a modification of the space carving algorithm [5].



(a) Detected points in camera 1. (b) Detected points in camera 2. (c) SIFT fundamental inliers

Figure 8: SIFT Experiment



(a) Detected points in camera 1. (b) Detected points in camera 2. (c) LGHD fundamental inliers

Figure 9: LGHD Experiment

The volume is determined by the following bounds: $x \in [-100], y \in [-37], z \in [-23]$. It is filled with voxels that whose sides have length 0.1.

Space Carving The implementation of Space Carving is written in the language *Julia* [16].
All cameras are positioned on the same side of the scene and looking in a similar general direction. Therefore it is sufficient here to only consider the single sweep version towards the negative x direction. A voxel is considered photoconsistent when all of the following conditions are satisfied.

- The mean image value is calculated in each backprojected voxel. The difference of means of all images needs to be smaller than 25.
- The standard distribution of image values is calculated for each backprojected voxel. The difference between all standard deviations needs to be smaller than 0.75.

The result is shown in Figure 10. In total there are 150 photoconsistent voxels found. Space Carving does detect pixel correspondences that belong to a similar region of the scene. When looking at the voxels in world space, they can be seen floating between two surface areas and not distributed along one. This likely is caused by the homogeneity of the color in the scene. It causes a voxel to be considered consistent, before the correct voxel is found. This can be seen as a representation of the "bulginess", inherent to the algorithm [5].

Contour Carving Based on the Space Carving and Reconstruction by Feature Correspondence experiments, a modified version of Space Carving, we call *Contour Carving*, is discussed and executed below.
We consider two objects S_1, S_2 that at every surface point have same lambertian radiance $rad_{S_1} \neq rad_{S_2}$. Let $c1$ be a camera for which S_1 partly occludes S_2, as shown in figure_figure_occlusion_. Then the image taken by $c1$ will contain an edge where both shapes

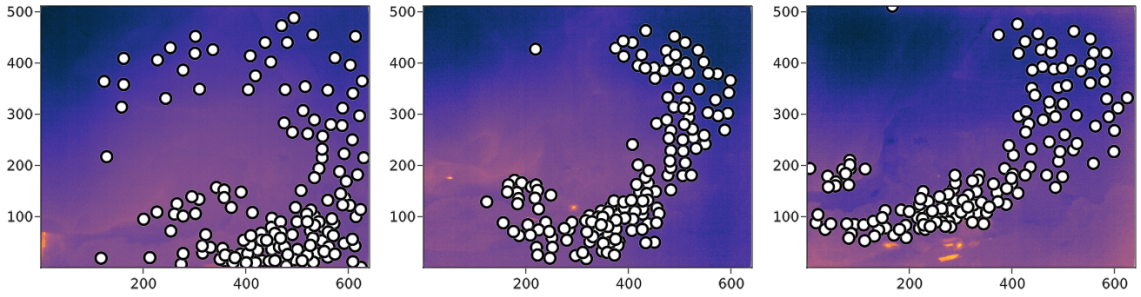


Figure 10: Projection of Photoconsistent Voxels into Cameras 1, 2 and 3 found by Space Carving.

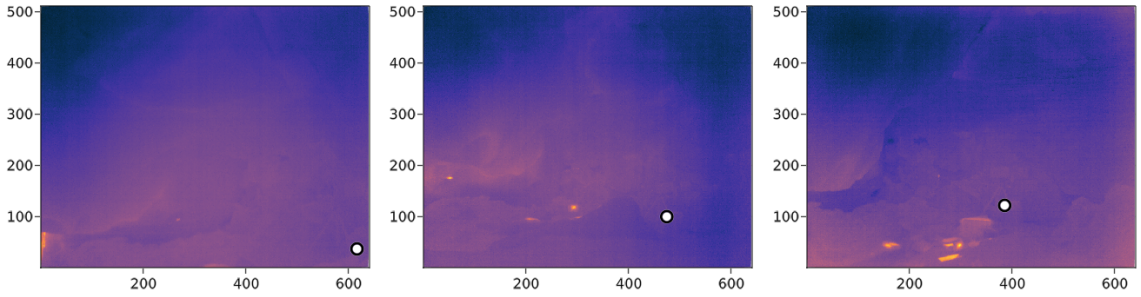


Figure 11: Projection of Photoconsistent Voxel(s) into Cameras 1, 2 and 3 found by Contour Carving.

overlap. Now we consider a second camera $c2$ which is located next to $c1$ however the position and facing direction is not exactly the same. S_1 and S_2 will cause an edge on the image of $c2$. However, the location on the surface that projects next to the edge will not be exactly the same as in camera $c1$. We observe, that this error gets smaller the more similar $c1$ and $c2$ get. So if we assume similar cameras we can associate the edges with each other and only get a small error.

We now apply above assumption to the space carving algorithm. The idea is to filter out low structure regions which don't have reliable information for feature description anyways. As discussed above, phase congruency detects edges that roughly correspond with the structure in the image a human would recognize [6]. An additional constraint for the consistency check gets added for this experiment: In the backprojected region of every consistent voxel needs to be at least one pixel with phase congruency $pc \geq 0.1$. This assures that only voxels are matched that are near an edge. The color information already considered in the consistency check for space carving then determines if the edge belongs to the same two objects. Phase Congruency is computed as described in section 5.2. Hypothetically, the image value criterion can be extended near edge regions. As stated above, we consider the objects that create the edge to be colored uniformly. If they create an edge, they have different colors. This allows us to rewrite the color criterion as having rad_{S_1} on one, and rad_{S_2} on the other side of the edge. However, the extended color consistency check is not considered in this experiment.

The contour carving version only considers voxels consistent which project near an edge and thus structure in the images. The observed results support this statement. Figure 11 shows the result of the experiment. There is only one matching point. The matching point is approximately at a position where it would be expected. However, based on this experiment the edge information does not seem to be sufficient to find corresponding points in the image in a useful quantity and precision.

6 Discussion

The result of the DMCP Experiment shows, that the proposed algorithm succeeds in estimating the pose of the camera. There still is some error in the registration, which can be explained by the fact that the initial correspondences were selected manually. Errors made in this step lead to errors in the pose calibration and the estimation of the transform. This result is in stark contrast to the other approaches.

Although SIFT, when used with the right threshold, does find many points, they are so unstable that only one match satisfies the epipolar constraint. This makes it extremely difficult to apply ICP-registration as the algorithm needs several well distributed points to work. ICP also relies on an initial transform that roughly aligns two point clouds. To compute an initial transformation a 3D point cloud descriptors might be employed. However it relies on a point cloud that is dense enough to describe the local shape. This is why the initial transformation can also not be computed from SIFT correspondences.
The LGHD approach does find more matches than the SIFT approach. However some matches, while satisfying the epipolar constraint, are not valid. This reduces the overall quality of the estimated point cloud. The resulting point cloud is very sparse still. Thus, this approach suffers from similar problems as described for SIFT. Advantages however are, that the detected points are very closely aligned to the structure boundaries which is a sign for more stable features. Another advantage of LGHD is that there are more valid matches. This shows that the phase congruency approach in general is preferable to the classical SIFT method.
In general, reconstruction by feature correspondence does not seem feasible.

Reconstruction by photoconsistent voxels shows better results than reconstruction by feature correspondence. The space carving algorithm finds many photoconsistent voxels. Nevertheless, does not estimate correct correspondences well enough to align camera and lidar space through ICP. Most pixel correspondences lay in the same general area of the scene. However, the contour carving version that tries to restrict correspondences to edge regions does not improve on space carving.

Based on the experiments, two things may be necessary for reconstruction robust enough to be able to align camera and lidar space. They seem especially important in a low structure thermal view of underground environments.

- A successful algorithm should take into account the reconstructed shape while estimating the pixel correspondences.
- A global continuous function could be incorporated to remove the ambiguity in textureless areas.

Finally, DMCP is the most successful method. It has the advantage that it uses a depth image from the backprojected lidar mesh for direct registration. It takes into account the lidar information and enables humans to annotate correct correspondences comparatively easily. It can be imagined that the depth map might also simplify the correspondence search for machines. This is a possible direction future work could reveal. Future work should also examine solutions that globally optimize the reconstructed shape directly in a continuous way.

7 Conclusion

This thesis approached the task of pose estimation for thermal cameras in a cave environment. It presents a general systematic for the task and applies several classical computer vision concepts techniques experimentally.
Methods that estimate feature matches to be used for later ICP-registration still present open challenges. The feature correspondence based experiments did not result in a point cloud dense and precise enough for ICP. The thoughts on contour carving did not improve on space carving.
The DMCP algorithm was presented as a main contribution. It directly registers lidar and calibrated cameras without a (lidar-) calibration target. The DMCP algorithm estimates the transformation from camera calibration space to the world space of the lidar mesh by using sparse correspondences to a backprojected depth map. With an average error of about 8 pixels, it is a feasible approach to the task.
The biggest drawback of the DMCP algorithm is the manual selection of corresponding points in a thermal image and a depth map. Shape based methods like space carving indicate that fully-automatic registration might be possible with algorithms that consider world space continuously and optimize an error globally.
To conclude, the semi-automatic DMCP solves the pose estimation task for subterranean thermal images, while leaving the challenge of indirect alignment by 3D reconstruction open for future work.

Appendices

A Numeric Results of DMCP

The numeric results of the DMCP algorithm are presented below. A is the resulting matrix that transforms points from calibration space to lidar space. $P_i, i \in \{1, 2, 3\}$ are the estimated projection matrices for the respective cameras in lidar coordinates. For completeness, the intrinsic matrix K is also provided. It is the same for all three cameras.

$$A = \begin{bmatrix} -0.7533 & 0.6353 & -0.1699 & -1.7994 \\ -0.6575 & -0.7332 & 0.1734 & 1.7945 \\ -0.0144 & 0.2424 & 0.9701 & 0.2003 \end{bmatrix}$$

$$P_1 = \begin{bmatrix} -97.4090 & 151.7501 & 15.2352 & -144.0903 \\ -40.0415 & 4.5967 & 167.2331 & -17.9918 \\ -0.2868 & -0.0080 & 0.0650 & 0.4852 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} -1.4306 & 116.1143 & 24.0421 & 196.9516 \\ -6.5108 & 2.4837 & 112.2417 & 196.9212 \\ -0.1568 & 0.0853 & 0.0725 & 0.5504 \end{bmatrix}$$

$$P_3 = \begin{bmatrix} 42.7547 & 132.1141 & 12.9215 & 258.0465 \\ -21.4681 & 26.4508 & 127.9226 & 93.1259 \\ -0.1455 & 0.1674 & 0.0436 & 0.4290 \end{bmatrix}$$

$$K = \begin{bmatrix} 526 & 0 & 320 \\ 0 & 526 & 256 \\ 0 & 0 & 1 \end{bmatrix}$$

The cameras rotation and translation can be extracted from the above projection matrices as the intrinsic matrix K is known. The translation then represents the estimated location of the cameras in lidar space. In the following, is demonstrated for camera 2. The relationships used are based on the pinhole projection model described in paragraph 2. Euclidean and homogeneous coordinates are distinguished precisely precisely.

Remove intrinsics from projection matrix. This results in the camera extrinsic matrix.

$$E_2 = K^{-1} \cdot P_2 = \begin{bmatrix} 0.0926 & 0.1689 & 0.0016 & 0.0396 \\ 0.0639 & -0.0368 & 0.1781 & 0.1065 \\ -0.1568 & 0.0853 & 0.0725 & 0.5504 \end{bmatrix}$$

The camera pose matrix is the inverse of the homogeneous embedding $\hat{E}_2$ of E_2.

$$\hat{\hat{E}}_2 = \hat{E}_2^{-1} = \begin{bmatrix} 2.4964 & 1.6915 & -4.2140 & 2.0402 \\ 4.5516 & -0.9743 & 2.2929 & -1.3384 \\ 0.0441 & 4.8060 & 1.9856 & -1.6065 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Because the value in the $4th$ row and column is equal to 1, we can directly read off the rotation and position.

$$R_2 = \begin{bmatrix} 2.4964 & 1.6915 & -4.2140 \\ 4.5516 & -0.9743 & 2.2929 \\ 0.0441 & 4.8060 & 1.9856 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} 2.0402 \\ -1.3384 \\ -1.6065 \end{bmatrix}$$

R_2 and T_2 now describe the orientation and location of camera 2 in lidar space as estimated by DMCP.

References

[1] August Ferdinand Möbius. *Der barycentrische Calcul*. Leipzig: Johann Ambrosius Barth Verlag, 1827.

[2] Berthold K. P. Horn. "Closed-form solution of absolute orientation using unit quaternions". In: 4.4 (Apr. 1987), p. 629. DOI: 10.1364/josaa.4.000629. URL: https://doi.org/10.1364/josaa.4.000629.

[3] Paul J Besl and Neil D McKay. "Method for registration of 3-D shapes". In: *Sensor fusion IV: control paradigms and data structures*. Vol. 1611. International Society for Optics and Photonics. 1992, pp. 586–606.

[4] Lisa Gottesfeld Brown. "A Survey of Image Registration Techniques". In: *ACM Comput. Surv.* 24.4 (1992), pp. 325–376. ISSN: 0360-0300. DOI: 10.1145/146370.146374. URL: https://doi.org/10.1145/146370.146374.

[5] Kiriakos N. Kutulakos and Steven M. Seitz. "A Theory of Shape by Space Carving". In: *International Journal of Computer Vision* 38.3 (2000), pp. 199–218. DOI: 10.1023/a:1008191222954. URL: https://doi.org/10.1023/a:1008191222954.

[6] Peter Kovesi. "Phase Congruency Detects Corners and Edges". In: *DICTA*. 2003.

[7] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Mar. 2004. DOI: 10.1017/cbo9780511811685. URL: https://doi.org/10.1017/cbo9780511811685.

[8] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: 60.2 (Nov. 2004), pp. 91–110.

[9] Claus Weitkamp, ed. *Lidar: Range-Resolved Optical Remote Sensing of the Atmosphere (Springer Series in Optical Sciences, 102)*. English. Hardcover. Springer, July 15, 2005, p. 476. ISBN: 978-0387400754.

[10] Kiana Hajebi and John S. Zelek. "Structure from Infrared Stereo Images". In: *2008 Canadian Conference on Computer and Robot Vision*. 2008, pp. 105–112. DOI: 10.1109/CRV.2008.9.

[11] H. Hirschmuller. "Stereo Processing by Semiglobal Matching and Mutual Information". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2 (Feb. 2008), pp. 328–341. DOI: 10.1109/tpami.2007.1166. URL: https://doi.org/10.1109/tpami.2007.1166.

[12] A. Vedaldi and B. Fulkerson. *VLFeat: An Open and Portable Library of Computer Vision Algorithms*. http://www.vlfeat.org/. 2008.

[13] Sanjeev J. Koppal. "Lambertian Reflectance". In: *Computer Vision*. Springer US, 2014, pp. 441–443. DOI: 10.1007/978-0-387-31439-6_534. URL: https://doi.org/10.1007/978-0-387-31439-6_534.

[14] Cristhian A. Aguilera, Angel D. Sappa, and Ricardo Toledo. "LGHD: A feature descriptor for matching across non-linear intensity variations". In: *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, Sept. 2015. DOI: 10.1109/icip.2015.7350783.

[15] Hussin K. Ragb and Vijayan K. Asari. "Histogram of oriented phase (HOP): a new descriptor based on phase congruency". In: ed. by Sos S. Agaian and Sabah A. Jassim. SPIE, May 2016. DOI: 10.1117/12.2225159.

[16] Jeff Bezanson et al. "Julia: A fresh approach to numerical computing". In: *SIAM Review* 59.1 (2017), pp. 65–98. DOI: 10.1137/141000671. URL: https://epubs.siam.org/doi/10.1137/141000671.

[17] Trong Phuc Truong et al. "Registration of RGB and Thermal Point Clouds Generated by Structure From Motion". In: IEEE, Oct. 2017. DOI: 10.1109/iccvw.2017.57.

[18] Liang Zheng, Yi Yang, and Qi Tian. "SIFT meets CNN: A decade survey of instance retrieval". In: *IEEE transactions on pattern analysis and machine intelligence* 40.5 (2017), pp. 1224–1244.

[19] Zhitao Fu et al. "HOMPC: A Local Feature Descriptor Based on the Combination of Magnitude and Phase Congruency Information for Multi-Sensor Remote Sensing Images". In: 10.8 (Aug. 2018), p. 1234. DOI: 10.3390/rs10081234.

[20] Shaharyar Ahmed Khan Tareen and Zahra Saleem. "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK". In: *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. Mar. 2018, pp. 1–10. DOI: 10.1109/ICOMET.2018.8346440.

[21] Eung-su Kim and Soon-Yong Park. "Extrinsic Calibration between Camera and LiDAR Sensors by Matching Multiple 3D Planes". In: *Sensors* 20.1 (Dec. 2019), p. 52. ISSN: 1424-8220. DOI: 10.3390/s20010052. URL: http://dx.doi.org/10.3390/s20010052.

[22] C. Bane Sullivan and Alexander Kaszynski. "PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK)". In: *Journal of Open Source Software* 4.37 (May 2019), p. 1450. DOI: 10.21105/joss.01450. URL: https://doi.org/10.21105/joss.01450.

[23] Ende Wang et al. "Infrared stripe correction algorithm based on wavelet decomposition and total variation-guided filtering". In: 16.1 (Dec. 2019). DOI: 10.1186/s41476-019-0123-2. URL: https://doi.org/10.1186/s41476-019-0123-2.

[24] Jaehyeon Kang and Nakju L. Doh. "Automatic targetless camera–LIDAR calibration by aligning edge with Gaussian mixture model". In: *Journal of Field Robotics* 37.1 (2020), pp. 158–179.

[25] Thejasvi Beleyur. "Theoretical and empirical investigations of echolocation in bat groups". PhD thesis. Konstanz: Universität Konstanz, 2021.

[26] Misha Urooj Khan et al. "A Comparative Survey of LiDAR-SLAM and LiDAR based Sensor Technologies". In: *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*. 2021, pp. 1–8. DOI: 10.1109/MAJICC53071.2021.9526266.

[27] *MATLAB version 9.10.0.1649659 (R2021a) Update 1*. The Mathworks, Inc. Natick, Massachusetts, 2021.

[28] P. D. Kovesi. *MATLAB and Octave Functions for Computer Vision and Image Processing*. URL: https://www.peterkovesi.com/matlabfns/ (visited on 02/22/2022).

[29] *ThermoViewer*. URL: https://thermalcapture.com/thermoviewer/ (visited on 02/22/2022).