Interpreting Thermal 3D Models of Indoor Environments for Energy Efficiency

**Girum G. Demisse · Dorit Borrmann ·
Andreas Nüchter**

Abstract In recent years 3D models of buildings are used in maintenance and inspection, preservation, and other building related applications. However, the usage of these models is limited because most models are pure representations with no or little associated semantics. In this paper we present a pipeline of techniques used for interior interpretation, object detection, and adding energy related semantics to windows of a 3D thermal model. A sequence of algorithms is presented for building the fundamental semantics of a 3D model. Among other things, these algorithms enable the system to differentiate between objects in a room and objects that are part of the room, e.g. floor, windows. Subsequently, the thermal information is used to construct a stochastic mathematical model– namely *Markov Random Field–* of the temperature distribution of the detected windows. As a result, the MAP(Maximum a *posteriori*) framework is used to further label the windows as either *open*, *closed* or *damaged* based upon their temperature distribution. The experimental results showed the robustness of the techniques. Furthermore, a strategy to optimize the free parameters is described, in cases where there is a sample training dataset.

1 Introduction

Efficiency in energy usage is a fundamental step in adopting green energy and conservation of natural resources: the European commission estimates the largest energy saving potential lies in residential ($\approx 27\%$) and commercial ($\approx 30\%$) buildings [Borrmann et al(2012)]. Among other factors uncontrolled air leakage, known as air infiltration, plays a significant role in energy consumption during heating seasons but also in geographical locations where air conditioning is a necessity.

Informatics VII : Robotics and Telematics
University of Würzburg, Germany
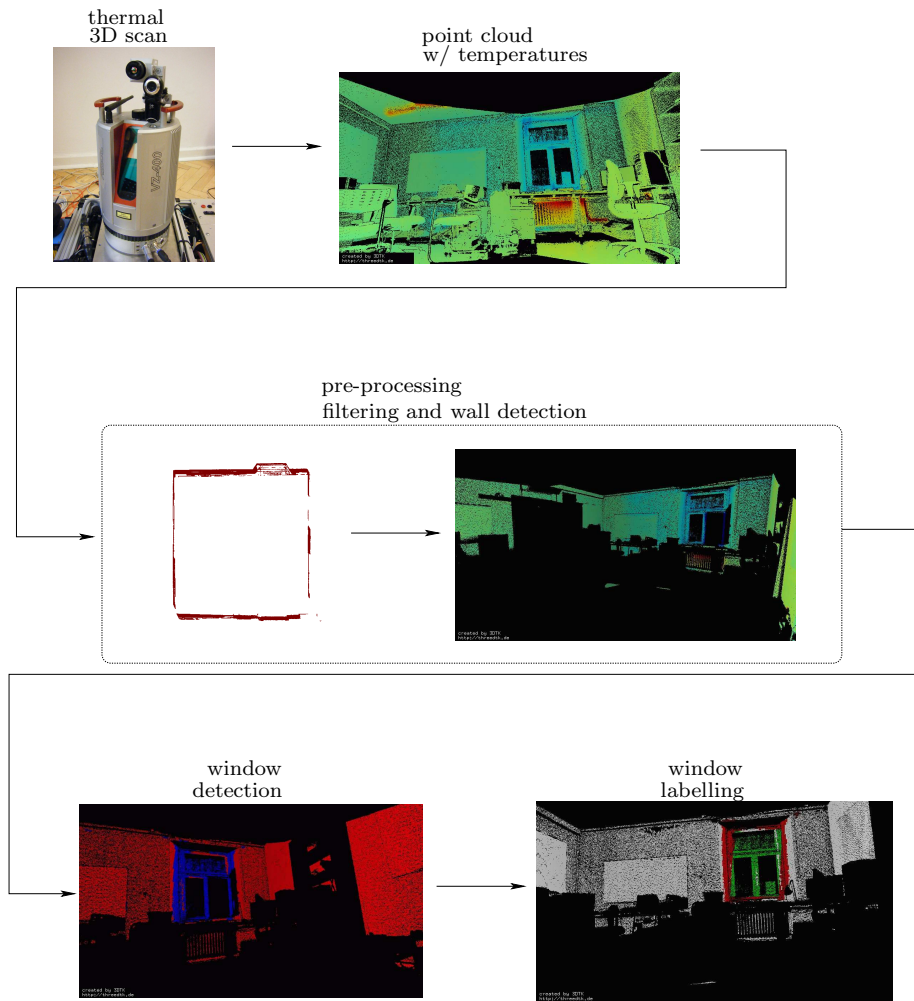E-mail: andreas@nuechti.de

Fig. 1: Overview of window Detection and labelling pipeline.

High rate of air infiltration is caused mostly by opened windows or doors which can easily be resolved by human intervention. However, other parts of a building that are poorly insulated are not easily detectable as open window or door. As a result, infrared thermometers are mainly used to detect faulty insulations in a labor intensive and time taking manner [Ham and Golparvar-Fard(2013a), Pedersen and Hellevang(2008)]. Alternatively, automating the process of air infiltration detection has a significant impact on efficiency, cost and effectiveness of the leakage detection and proofing process. Consequently, there is an ongoing research in automated 3D model creation for energy efficiency [Laguela et al(2014), Ham and Golparvar-Fard(2013b)].

Motivated by the economic and environmental impact we contribute to the efforts of fully automating the energy leakage detection process. Building on results obtained in [Borrmann et al(2012)], where a method for acquiring a 3D thermal model of a building is presented, we present a sequential pipeline of algorithms for 3D scene understanding and temperature distribution modelling as given in Fig. 1. Particularly, the temperature distribution is used to model the state of a window as either opened, closed, or damaged, i.e., not properly insulated. After reviewing related work we describe our mobile robot and the thermal camera vs. 3D laser scanner calibration to make the paper self-contained. Then we define and formalize the problem mathematically in Section 4. Our solution pipeline uses probabilistic model and pre-processing of a 3D point cloud and is presented in Section 5 and 6. Finally, experimental results are presented in Section 7. Section 8 concludes the paper.

Throughout the paper we use a data set acquired at Jacobs University Bremen, Germany. The buildings feature a uniform type of windows as given in figure 2. Thermography however is preferred to be performed in interiors. To detect energy related flaws a difference of 15 Kelvin is necessary between indoor and outdoor temperature to come to significant conclusions. It is desired that the weather conditions remain stable over a longer period of time, making the morning hours in the winter months ideal. Keeping stable conditions is easier to achieve for indoor thermography. The analysis of back-ventilated walls and roofs is only possible from indoors. Thermal bridges at exterior walls and interior walls connecting heated and unheated rooms, pillars that interrupt the thermal insulation of a building, air leaks at windows and doors and the moisture penetration at basement walls are the common applications for indoor thermography, that focus on energy efficiency in existing buildings [Fouad and Richter(2012)].

2 Related work

According to Xiong and Huber, creating a 3D model of an indoor environment has notable advantages in maintenance, management, and architectural renovation of buildings [Xiong and Huber(2010)]. The traditional approach to create a 3D model is based on CAD (Computer-aided design) tools and manual measurements, despite the consequential high cost and lengthy time consumption. However, recent technological advances in laser scanning technology prompted the full automation of 3D model creation [Adan and Huber(2011), Surmann et al(2003)]. 3D model creation is still a fairly complex task with challenges at different levels. These challenges can be categorized in two; *lower level* problems and *higher level* problems. Lower level problems are successfully explored in the computer vision community. They include the mere acquisition of data as well as basic post-processing steps. In 3D point cloud processing the acquisition of accurate data is solved by technological means. Computer vision methods are widely used to solve fundamental problems such as registration [Besl and McKay(1992)], i.e., solving simultaneous localization and mapping [Borrmann et al(2008)], and representation, i.e., octrees [Hornung et al(2013)] and range images. The addition of semantics belongs to the second group of problems. Semantics range from primitive shape detection to higher level knowledge inference. Several shape detection methods have been proposed in the computer vision community that can cope with un-

Fig. 2: Above: The campus center of the Jacobs University as a 3D point cloud with thermal information and filtered vegetation [Elseberg et al(2011)]. Below: Photo of eight windows (left) and the corresponding infrared image (right). In the thermal image heat losses are clearly detectable. In the top row the windows show different temperature distributions. The first window on the left belongs to an empty office which is not heated. The top part if the second window is open. The third window is closed and has proper insulation. The fourth window shows unusually high temperatures. This was caused by a partially detached rubber insulation strip that allowed the air to flow through.

certainties and clutter in the data sets [Holz et al(2011), Schnabel et al(2008)]. Additionally, the current trend in 3D point cloud interpretation is to infer higher level knowledge [Pangercic et al(2012)].

Building thermal 3D models of environments received some attention recently. Ham and Golparvar-Fard model and evaluate thermal models of building exteriors and the energy performance of buildings [Ham and Golparvar-Fard(2013a)]. However, to the best of our knowledge, there has not been any work done in automatic temperature analysis for the understanding of an object state.
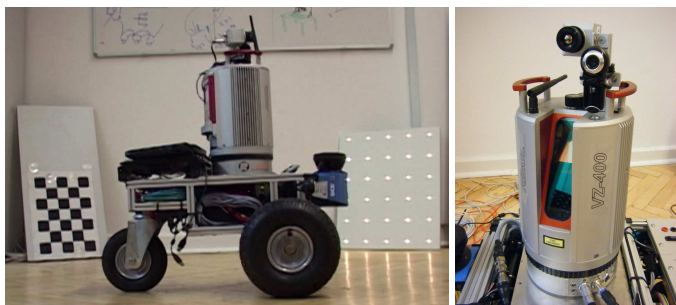
Fig. 3: The robot Irma3D, with a 3D laser scanner, a thermal camera and a webcam.

3 Automatic Acquisition of Thermal 3D Models

Thermal imaging is state of the art in recording energy related issues [FLIR Systems AB(2011), Ham and Golparvar-Fard(2013b), Laguela et al(2014)], while terrestrial laser scanning has been used for years to create 3D models [Adan and Huber(2011), Xiong and Huber(2010), Vosselman and (Eds.)(2010)]. The combination of these two yields a 3D model that contains precise temperature information including the dimensions of heat and air leaks. To achieve valid results some general rules, mostly concerning the weather conditions, have to be taken into account when performing thermography. We pay attention to these rules in all our experiments. Most importantly, to measure a valid, noise-reduced thermogram there has to be a temperature difference between indoor and outdoor of at least $15°$. Other error sources such as sun light, wind and rain, clear sky were minimized as well.

Experimental Setup and Data Acquisition. The setup for simultaneous acquisition of 3D laser scan data and thermal images is the robot Irma3D (see Fig. 3). Irma3D is built of a Volksbot® RT-3 chassis. Its main sensor is a Riegl® VZ®-400 laser scanner from terrestrial laser scanning. A thermal camera is mounted on top of the scanner. The optris® PI160 thermal camera has an image resolution of 160×120 pixels and a thermal resolution of $0.1°$ C. It acquires images at a frame rate of 120 Hz and with an accuracy of $2°$ C with a field of view of approximately $40° \times 64°$. The laser scanner acquires data with a field of view of $360° \times 100°$. To achieve the full horizontal field of view the scanner head rotates around the vertical scanner axis when acquiring the data. We take advantage of this feature when acquiring image data. Since the cameras are mounted on top of the scanner, they are also rotated. We acquire 10 images per camera during one scanning process to cover the full $360°$.

After acquiring the 3D data it has to be merged with the image information. This processing consists of five steps that will be explained in this section.

Intrinsic Calibration of Thermal and Optical Camera. Each sensor perceives the world in its own local coordinate system. To join the perceived information we need the specific parameters of these coordinate systems. Each camera has unique parameters that define how a point (X, Y, Z) in world coordinates is projected

onto the image plane. These parameters are calculated through a process known as geometric camera calibration. Given the focal length (f_x, f_y) of the camera and the camera center (c_x, c_y) image coordinates (x, y) are calculated as:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix}. \tag{1}$$

Given the radial distortion coefficients k_1, k_2, k_3 and the tangential distortion coefficients p_1, p_2 and $r = \sqrt{x^2 + y^2}$ the corrected image points (x_c, y_c) are calculated as

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \begin{pmatrix} x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 y + p_2(r^2 + 2x^2) \\ y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2y^2) + 2p_2 x \end{pmatrix} \tag{2}$$

To determine the parameters of optical cameras, chessboard patterns are commonly used because the corners are reliably detectable in the images. A number of images showing a chessboard pattern with known number and size of squares are recorded. In each image the internal corners of the pattern are detected and the known distance between those in world coordinates allows to formulate equations (1) and (2) as a non-linear least squares problem and solve for the calibration parameters [Bradski and Kaehler(2008)].

For low resolution thermal cameras a chessboard pattern is error-prone even after heating it with an infrared lamp. For pixels that cover the edge of the squares the temperature is averaged over the black and white parts thus blurring the edges. [Luhmann et al(2010)] have explored the calibration procedure using different types of thermal cameras. Generally an object with a unique pattern having distinct targets is used which eases labeling and increases accuracy of the calibration process. The points are actively or passively heated. In case of passive heating different material causes the pattern to show up. [Luhmann et al(2010)] developed a pattern consisting of targets of self-adhesive foil on an aluminum plate. While the targets emit radiation related to their own temperature the reflective metal surface reflects the cold temperature of space thus leading to an enormous contrast in temperature. Unfortunately this concept is not applicable for the co-calibration of the thermal camera and a laser scanner as it is very difficult to position the board in a way that the sky is reflected without occlusions and the board is completely visible in the laser scan. Instead we suggest a pattern with clearly defined heat sources such as small light bulbs as it shows up nicely in thermal images.

Fig. 3 shows our pattern in the background. It is composed of 30 tiny 12 Volt lamps, each with a glass-bulb diameter of 4 mm. The overall size of the board is 500 mm (width) × 570 mm (height). Identifying the heat sources in the image enables us to perform intrinsic calibration in the same way as for optical cameras. To detect the light bulbs in the thermal image a thresholding procedure is applied to create a binary image showing regions of high temperature. A further thresholding step discards effectively all regions that are too big or too small. If the remaining number of regions is equal to the number of light bulbs in the pattern the regions are sorted according to the pattern to allow for easy determination of correspondences. To calculate the exact center of the features, the mean is calculated by weighing all the pixels in the region by its temperature value.

Extrinsic Calibration – Cameras and Laser Scanner. After calculating the internal parameters of the cameras we need to align the camera images with the scanner coordinate system, i.e., extrinsic calibration. The three rotation and three translation parameters are known as the extrinsic camera parameters and define the geometric relation between camera and laser scanner. Once all the points are in the camera coordinate system, the projection to the image can be defined up to an factor s using equation (3) [Bradski and Kaehler(2008)]:

$$
s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3}
$$

Suppose there are n images of the calibration pattern and m planar points on the pattern considering the distortions as independent and identically distributed noise then the maximum likelihood estimate of the transformation between the scanner and camera coordinate system is obtained by minimizing

$$
\sum_{i=1}^{n} \sum_{j=1}^{m} ||\mathbf{p}_{ij} - \hat{\mathbf{p}}(\mathbf{A}, \mathbf{D}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{P}_j)||^2 \tag{4}
$$

where $\mathbf{A}$ is the intrinsic matrix, $\mathbf{R}_i$ the rotation matrix, $\mathbf{t}_i$ the translation vector, and $\mathbf{D}$ the distortion parameters. $\hat{\mathbf{p}}(\mathbf{A}, \mathbf{D}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{P}_j)$ defines the projection of point $\mathbf{P}_j$ in image i, according to equations (3) and (2). This approach assumes that we have a number of points that are identifiable in both the laser scan and the image. For this purpose we attach the calibration pattern onto a board. For the optical camera this is a printed chessboard pattern and for the thermal camera light bulbs arranged in a regular grid pattern. The calibration patterns are depicted in the background of Fig. 3 The position of the points of these patterns are known. Algorithm 1 detects the points in a laser scan.

Algorithm 1: Calibration pattern detection in a laser scan.

  Require: point cloud, specification of calibration pattern
   1: discard points outside the area of the expected board
   2: find the most prominent plane using RANSAC (RANdom SAmple Consensus) [Fischler and Bolles(1981)]
   3: project a generated plane model into the center of the detected plane
   4: use ICP (Iterative Closest Point) algorithm [Besl and McKay(1992)] to fit the plane model to the data points
   5: **if** each point from the plane model has a corresponding point in the point cloud **then**
   6:     **return** position of the light bulbs according to ICP result
   7: **end if**

3D to 2D Projection and Color Mapping. During the data acquisition phase laser scans and images are acquired simultaneously. After determining the relations between scanner and cameras in the calibration step this relation is used directly to assign temperature and color values to the point cloud.

Projection/Occlusion/Resolution Errors. Due to the different fields of view the sensors see different parts of the world. An area that is visible for one sensor might be occluded for the other sensor. When mapping the thermal information to the point cloud this causes wrong correspondences and therefore faulty assigned values. This impact is amplifed by the low resolution of the thermal camera. With only 120 by 160 pixels per image each pixel corresponds to many 3D points seen by the laser scanner leading to errors at edges. Consequently small calibration inaccuracies have a large impact on the results. To solve this problem we perform a ray tracing procedure that checks whether a point in the point cloud can be seen by the camera. We connect the point $\mathbf{P}$ and the camera position $\mathbf{C}$ with a straight line PC and select all points with a distance less than a threshold t to PC, i.e., all points $\mathbf{O}_i$ for which

$$|\mathbf{P} - \mathbf{O}_i|^2 - \frac{|(\mathbf{P} - \mathbf{O}_i) \cdot (\mathbf{P} - \mathbf{C})|^2}{|\mathbf{P} - \mathbf{C}|^2} < t^2 \tag{5}$$

holds true. If any point $\mathbf{O}_i$ lies between $\mathbf{P}$ and $\mathbf{C}$, $\mathbf{P}$ is not visible from the camera and therefore is discarded. The threshold t accounts for small inaccuracies in the calibration and the low resolution of the camera simultaneously. To speed up the checking procedure the points are organized in a kd-tree data structure. With a quick check all voxels that are not traversed by the ray are immediately discarded and therefore all the points within are ignored.

4 Problem Definition

Inferring higher level knowledge about the property of an object can be seen as two problems that are highly related. First, the detection of objects that belong to a certain class. This is a problem where the emphasis is on understanding and modelling of time-invariant properties of a certain class of objects, e.g. all windows are made of glass, so that the properties are used to recognize an object of that class. Second, the inference about the *object*, rather than the class, using specific knowledge. In the second case, the problem is recognizing properties of an object that are observed in a certain time frame under a certain condition; which, of course, gives information about the objects state rather than the class it belongs to. Consequently, each of the above problems is usually solved separately and sequentially. In fact, the solution space of the first problem is the domain/problem space of the second problem. In this paper we will be dealing with window detection and labelling, i.e, assessing windows either as open ($\mathbf{O}$), closed ($\mathbf{C}$) or damaged ($\mathbf{D}$), i.e., a window without the proper insulation. Window detection belongs to the first group of problems and will be performed first followed by window labelling that belongs to the second group of problems.

Although window detection from a point cloud representation of a room full of objects is a challenging task we have designed a sequential pre-processing modules which essentially reduce window detection to the estimation of a mapping function $f(\cdot)$. Assuming we have successfully identified 3D points that represent a window, the 3D points are considered as random variables taking a temperature value from $\mathbb{R}$. Thus, the labelling of windows as closed, opened or damaged is formulated as a probability distribution modelling problem. In general, probabilistic models for object recognition are categorized either as *generative models*

or *discriminative models*. The former one attempts to model the *joint* probability distribution $P(X, Y)$ between the data denoted by X and the label denoted by Y. Alternatively, the discriminative model approach is to model the *posterior* probability $P(Y|X)$ directly from the data. A discriminative model is widely believed to be the better modelling technique with better predictive ability [Jordan(2002)]. Consequently, we have chosen the discriminative modelling approach, and thus the *posterior* probability of a label $y \in \{\mathbf{C}, \mathbf{O}, \mathbf{D}\}$, is modelled as:

$$t_s = f(x), \tag{6}$$

$$P(y|t_s) = \frac{p(t_s|y)P(y)}{p(t_s)}, \tag{7}$$

where x is the fully registered thermal 3D point cloud model of the room, $f(\cdot)$ is a function that takes this model as an input and outputs the temperature distribution of the detected window, $t_s \in \mathbb{R}^n$, where n is the number of 3D points representing the window. Since $p(t_s)$ is exactly the same for all the labels it has no effect on the label specific *posterior*, and thus is ignored. Additionally, the probability of a window being closed is assumed to be exactly the same as being open or being damaged. In fact, this might not be true but we have no prior information to assume otherwise. Therefore, the modelling task is:

$$P(y|t_s) \approx p(t_s|y) \tag{8}$$

In summary, labelling a window as open, closed or damaged is formulated as estimating the conditional probability distribution of every label, followed by a decision rule, where the decision will be based on MAP (Maximum *a posteriori*) to minimize the expected error [Bishop(2006)]. MAP is summarized as:

$$\hat{y} = \arg \max_{y_i \in Y} P(y = y_i|t_s) \tag{9}$$

where $\hat{y}$ is the final label assigned to the temperature distribution of a window $t \in \mathbb{R}^n$, and $Y = \{\mathbf{C}, \mathbf{O}, \mathbf{D}\}$. In the following sections a solution to the problem formulation given in Eq. (7) is presented, respectively.

5 Window Detection

Window detection is a difficult task with many associated problems. In this paper a simple but effective detection technique, exploiting the thermal camera, is presented. Intuitively, the core idea is the difference in material property, i.e., all windows are made of glass and walls are made of some other material. This means, given a thermal 3D point cloud of a room that contains only windows and walls, there is a thermal conductivity difference between the wall material and the window material. And thus, there will always be a temperature difference which can be used to recognize one from the other. However, a 3D point cloud representation of a room contains other objects and clutter. As a result, a mandatory preprocessing has to be done to detect and remove these other objects from the scene.
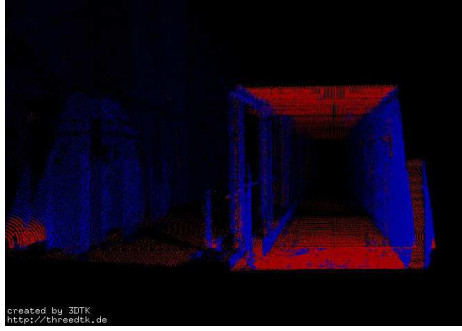
Fig. 4: Blue color represents the walls, and the red is for floor and ceiling; constraint is set to $\geq 80^0$

5.1 Pre-processing

Filtering objects that are inside a room and points that are scanned through windows (see Fig. 5) is a challenging task that is simplified with practical and realistic assumptions. The assumptions taken are:

– A room has a rectangular shape.
– The scanner is located inside the room.
– The walls and windows are not *completely* occluded, i.e., some part of the wall is always visible.
– The thermal 3D point clouds are registered to a single co-ordinate system.

Assuming the above conditions are true, a sequential procedure is proposed for filtering out objects from the scene, such that the point cloud consists only of walls and windows.

5.1.1 From 3D to 2D

Assuming that windows are located on the walls and not on the ceiling, neither the height of the room nor points representing the floor and ceiling are important for window detection. Hence, points representing the floor and ceiling need to be filtered out from the scan. Floors and the ceilings are, again, almost always parallel to each other and perpendicular to the walls. Thus, the normal vector of a 3D point representing the ceiling is parallel to a wall. Given the above we conclude that a 3D point is representing the floor or ceiling if its normal vector is perpendicular to the $x - y$-plane, or parallel to the $x - z$-plane, where z is the vertical axis. Let $\mathbf{a} = (1, 0, 0)$ be a vector on the x-axis, and $\mathbf{n}_i$ be the normal vector of the ith point, then for all points on the floor or on the ceiling $80° \leq \theta \leq 100°$ holds true in:

$$\mathbf{a} \cdot \mathbf{n}_i = \|\mathbf{a}\|\|\mathbf{n}_i\| \cos\theta \tag{10}$$

We set the threshold to $10°$, due to noise and inaccuracy in calculating normals.

Inspired by the work of Xiong and Huber [Xiong and Huber(2010)] our software further simplifies the 3D representation of the room to a 2D representation, since

the height of the room is not really needed in wall detection. Thus, the 3D points are transformed to a 2D plane as follows:

$$\begin{pmatrix} \mathbf{p}' \\ 0 \end{pmatrix} = \mathbf{p} - (\mathbf{p} \cdot \mathbf{n})\mathbf{n} \tag{11}$$

where $\mathbf{n}$ is the normal vector of the projection plane, $\mathbf{p}' \in \mathbb{R}^2$ is the projected point, and $\mathbf{p} \in \mathbb{R}^3$, is the point to be projected. Now that the room is projected onto a 2D plane walls are represented with lines instead of planes (cf. Fig. 5).

5.1.2 Wall Detection

The projection of the room to a 2D space enables us to work with lines as walls instead of planes. We have assumed that the room has a rectangular shape, that the origin of the scan is inside the room, and that the walls are not completely occluded. Hence, we can conclude that the $x - y$-axis of the 2D plane, which has its origin at the scanner location, intersects with each wall line after θ rotation, where $0° \leq \theta < 360°$. The intersection is independent of the orientation and position of the scanner as long as it is in the room. The only exception is when the axes intersect with the corners.

As a line equation is defined by two points that lie on the line we need to detect two points on each wall (line) to detect all the walls of the room. The *first* four points– since there are four walls– are determined by selecting the farthest point on the $\pm$ x-axis and $\pm$ y-axis. To determine the *second* set of points the registered scans are rotated by a given angle θ_2. Again the farthest point on each $\pm$ $x - y$-axes are selected. Rotating the selected points back by $-\theta_2$ yields the second set of points needed for defining the four line equations. Although θ_2 can be selected arbitrarily its size has a significant impact on the convergence speed of the algorithm. For larger θ_2 it is very likely that the rotation passes the corner causing the pairs to be on different walls. For small θ_2 the distance between the two points of a pair is small influencing the impact of noise in the data.

The algorithm fails in cases where the farthest point on one of the axes is not a wall point, e.g. due to occlusions, or due to the presence of noise, i.e., points outside of the room that are scanned through windows and doors. Further problems occur when one of the intersecting points lies in a corner of the room. To ensure an accurate and robust wall detection the following conditions are introduced:

- Each detected line has to be perpendicular to two lines, the ones it is intersecting with, and parallel to the other, the one it is not intersecting with. This is a hard constraint that has to be fulfilled in order to detect the walls with a reasonable accuracy; in other words, the slopes S_1 and S_2 of two intersecting lines should satisfy $S_1 = -1/S_2$.
- To avoid the detection of outliers as walls there should be a considerable number of points at all four intersection points of the four detected lines. This is more of a soft constraint, especially, in a heavily occluded scan. Consequently, this constraint is mainly used to measure the confidence level, i.e., $error = 1 - confidence$.

The confidence level is quantified by counting the intersection points of the detected four lines that have a considerable number of points on and around them,

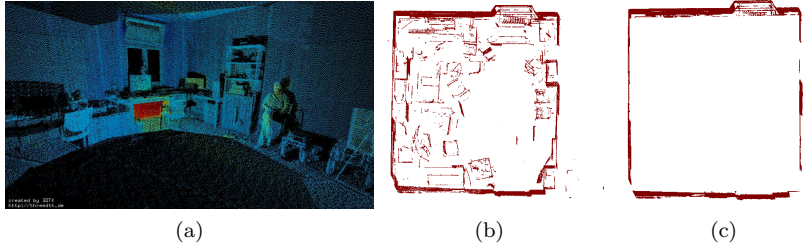(a)   (b)   (c)

Fig. 5: (a) A 3D thermal model of the Automation Lab at Jacobs University Bremen. (b) 2D projection of the room with the x-y axis colored in red. (c) The detected outer rectangular shape, or walls of the room.

e.g., if of the four intersection points three have a significant number of points on and around them then the confidence is 3/4 or 75%. The threshold for determining the significance of the number of points depends on the density of the scan. If the above two constraints are not satisfied the whole scan is rotated by a certain angle θ_1 and the line detection process starts all over again– the iteration continues until lines are detected that satisfy the constraints reasonably. The procedure is outlined in Algorithm 2. After a successful detection of the lines representing the walls, all points that are not near the walls, with in a certain threshold, are removed. An example is given in Fig. 5.

Algorithm 2: Wall Detecting Algorithm

Data: Registered and pre-processed scan
Result: Equation of four lines
initialization:
search = true
// ε_p threshold for perpendicularity
// θ_1, θ_2 step angle for line detection
while *search* **do**
    calculate the first axis-line intersections
    rotate scan by θ_2
    calculate the second axis-line intersections
    rotate scan by $-\theta_2$
    calculate the lines equation
    if $|S_1 + \frac{1}{S_2}| < \varepsilon_p$ *and* $|S_3 + \frac{1}{S_4}| < \varepsilon_p$ **then**
        count = # of corner points where significant # of points are found
        confidence = count/4
        search = false
    else
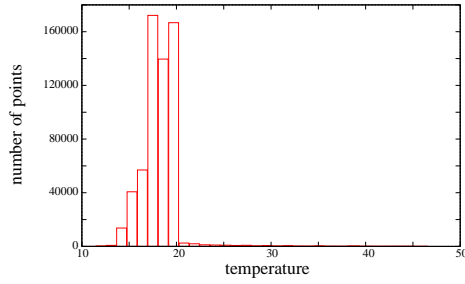   rotate scan by θ_1

Fig. 6: Typical temperature distribution of a 3D scan with walls and windows only.

5.2 Window Detection

The result of the pre-processing step is a 3D model of the walls of the room. Now the remaining point cloud is dominated by points from the wall which means most points have an almost similar temperature value. However, 3D points of a window and points around it show a considerable temperature difference from the wall points (cf. Fig. 6). Thus, the window detection technique aims to exploit these temperature differences as a main feature.

Since there are significantly more 3D points that represent the walls than those that represent the window area a typical temperature distribution in a room takes a bell curved shape (cf. Fig. 6). The rare ends of this temperature distribution correspond to temperature peaks. On the lower end these are objects that are cooler than the room temperature, i.e., mostly windows. The upper end represents hotter objects that are close to the wall, e.g., computers or heaters. Thus, filtering points with a constant threshold that is dependent on the standard deviation and mean of the temperature distribution enables us to detect points with uncommon temperature values, regardless of the the temperature distribution's peakedness or flatness. Although it must be noted that for sensitive thresholding one has to consider the possible asymmetry of the distribution, e.g., heaters might be turned off. Furthermore, in the warm season, when air conditioning is used, windows will contribute to the upper end of the temperature scale. The thresholding constant that divides the points into warm and cold points depending on the distribution is given as follows:

$$temp_{\mathrm{div}} = \frac{1}{N} \sum_{j=1}^{N} t_j \pm \sqrt{\frac{1}{N} \sum_{j=1}^{N} (t_j - \mu)^2} \qquad (12)$$

where t_j represent the temperature values of the room with walls, windows and other nearby objects, and μ is the mean temperature value.

Using the temperature dividing threshold $temp_{\mathrm{div}}$ we detect points with lower temperature values and conclude that these points represent windows. In the next step the potential window points are clustered according to their spatial distance from each other. The clustering is done based on a simplified version of k-means clustering, i.e., the clusters emphasize compactness and connectedness. See Algorithm 3 for summary. A final filtering procedure removes clusters with a small number of points.

Algorithm 3: Clustering Algorithm

Data: Potential 3D window points P
Result: Set of clusters S
S = empty
set distance threshold ε_d
for $\forall p_i \in P$ **do**
    pointadded = false
    for $\forall S_a \in S$ **do**
        for $\forall p_j \in S_a$ **do**
            dist = $\|\mathbf{p}_i - \mathbf{p}_j\|$
            if $dist \leq \varepsilon_d$ **then**
                add p_i to cluster S_a
                pointadded = true
                break;

        if *pointadded* **then**
            break;
        pointadded = false
    if *!pointadded* **then**
        create new cluster S_{new}
        add p_i to S_{new}
        add S_{new} to the cluster set S
    remove p_i from P

Each of the remaining clusters is processed individually in case there is more than one window in the room. We approximately determine the width and height of each window from the cluster. This has the major advantage that we can control the number of points on the window. The boundaries of the window are approximated by first determining the plane that the window lies on by removing the axis with the smallest variance. Second, we select the extreme $\pm$ of each axis, i.e., the distance from the respective component of the mean vector of the clusters. Finally, the extreme points are used to determine the boundary points. To achieve an identical number of points on each detected window we use an octree based sub-sampling (see [Elseberg et al(2012)]). The procedure takes into account the size of the window by creating an octree with a variable minimum voxel size and taking only one point from each voxel.

6 Modelling state of a window

In this paper a window is considered to be in three distinctive states (labels), henceforth will be used alternatingly. The states are: closed(**C**), opened(**O**), or damaged(**D**) – a damaged window refers to a window without the proper heat insulation. In this section, MRF (Markov random field) is presented as a mathematical tool to model the probability distribution of each window's states, given the 3D thermal point cloud of the window.

6.1 Markov random field

First introduced by [Besag and Moran(1975)] the Markov Random Field (MRF) is mainly used to express the statistical dependencies between several random

variables arranged in some form of spatial configuration or *graph*. MRF simplifies the joint distribution modelling by exploiting the Markovian property of the graph [Clifford(1990)].

Meanwhile, the *Hammersley-Clifford* theorem [Besag and Moran(1975)] showed that a joint probability distribution of several random variables $\{z_1, \cdots, z_n\}$ that can be factorized as $P(z_1, \cdots, z_n) = \prod_{j=1}^{n} \phi_j(\psi_j)$, where ϕ_j is any real valued function defined for each clique set, is equivalent with MRF [Clifford(1990)]. Furthermore, since MRF assumes that every joint realization of the random variables has a strictly positive probability value– this is called the positivity condition– the factorized distribution can be expressed using exponential form, called Boltzmann (Gibbs) distribution. Boltzmann distribution is defined as:

$$p(t; \theta) = \frac{1}{Z(\theta)} \exp(\frac{-E(t, \theta)}{T}) \tag{13}$$

where Z is the normalization constant, T is a controlling constant, t is a certain realization of the random variables, and $E(\cdot)$ is a system, i.e. the system being modelled, specific function called energy function. And finally, θ is a set of free constant parameters that will be discussed later.

6.1.1 Boltzmann distribution

The Boltzmann distribution is at the heart of MRF modelling; hence, in this subsection, a brief discussion about the mathematical properties of the Boltzmann distribution is presented. There are two independent variables $E(\cdot)$ and T and a single dependent variable $p(\cdot)$ in Eq. (13). Consequently, understanding the exact relationship between each independent variable and the probability distribution is crucial.

The first and perhaps the most important property is the relationship between the energy function and the distribution. Since the exponential function is taking the negative of the energy function as an input, and given the nature of exponential functions, it is straight forward to see the inverse variation of the energy function and the distribution. And thus, assuming $E(\cdot) > 0$, the relationship can be expressed as $p(\cdot) \propto \frac{1}{E(\cdot)}$. Intuitively, this means the Boltzmann distribution favours states with small energy value– in terms of modelling, it means all instances of a class should always minimize the energy function of the class's Boltzmann distribution, more accurately, should minimize the energy function better than other class's instance. The second property on the other hand, is between the controlling constant T and the distribution; *as T gets small the standard deviation of the distribution gets small as well*. In essence, $T \propto \sigma$, where σ is the standard deviation.

In the next subsection, a discussion is presented about the formulation of energy functions that favour a distinctive feature of window's states.

6.2 Energy functions

Apparently, the temperature distribution of a closed window–a window that is neither damaged nor opened–exhibits a very small variance regardless of the peak temperature. On the contrary, the temperature distribution of opened window
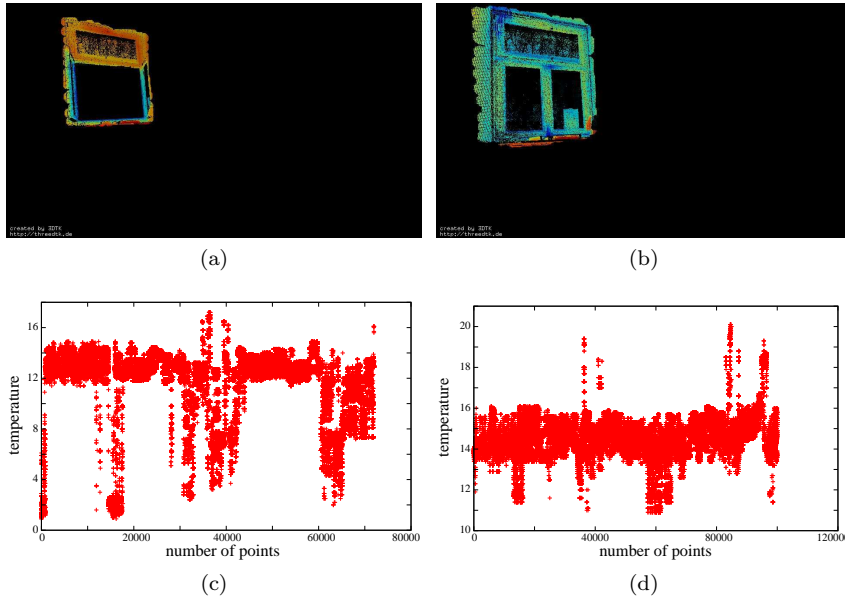
(a)

(b)

(c)

(d)

Fig. 7: (a) Open window colored according to the thermal distribution. (b) Closed window colored according to the thermal distribution. (c)+(d): the temperature distribution of each window, respectively.

has a much higher variance than closed window. This has shown to be a very robust feature, i.e., invariant to peak temperature value, to distinguish one from the other. Damaged windows, however, are particularly detectable because of their non-smooth temperature distribution. Intuitively, the non-smoothness in temperature value is the main feature of a leaky window. Therefore, non-smoothness in temperature distribution is the emphasized feature in the energy function of a damaged window. However, it must be noted that the temperature distribution of both opened and closed windows are smooth, i.e., small temperature difference between neighbouring points, as opposed to the distribution of damaged windows. These characteristics of the different window states were derived from thermal images of windows. Examples of different windows are shown in Fig. 2.

On the other hand, energy functions are ought to be designed such that the function is close to the minimum for strong cases of the designated feature and reaches its minimum for extreme cases, respectively. Additionally, since the features of each state should be distinctive they must be different from each other and the design of the energy function is state specific. The following equations, thus, are designed as an energy function for each state.

$$E_C(t_s, \theta_c) = \alpha_{1c} \sum_{j=1}^{N} \frac{(t_j - \mu)^2}{N} + \alpha_{2c} \sum_{j=1}^{N} \frac{d(x_j, K)}{N} \tag{14a}$$

$$E_O(t_s, \theta_o) = \alpha_{1o} \frac{N}{\sum_{j=1}^{N}(t_j - \mu)^2} + \alpha_{2o} \sum_{j=1}^{N} \frac{d(x_j, K)}{N} \tag{14b}$$

$$E_D(t_s, \theta_d) = \alpha \frac{N}{\sum_{j=1}^{N} d(x_j, K)} \tag{14c}$$

where $\theta_c = (\alpha_{1c}, \alpha_{2c})$, $\theta_o = (\alpha_{1o}, \alpha_{2o})$, $\theta_d = (\alpha)$ are weighting constants for each term of the energy functions. N is the number of random variables (3D points) representing the window. d is a function that calculates the average temperature difference in a K neighbouring points, with respect to x_j's spatial location. And finally, t_j is the temperature value of the jth point x_j of the window and μ the mean temperature value. The first term encourages small variance in case of a closed window's energy function Eq. (14a), and a high variance in case of open window's energy function Eq. (14b). The second term in both cases, open and closed window, emphasizes smoothness. However, the damaged window's energy function Eq. (14c) encourages non-smooth temperature distribution (cf. Fig 8).

The energy functions, however, are defined on a very high dimensional space and computing the normalization constant analytically is mathematically complex and therefore not done in practice. Hence, it has to be approximated with an appropriate numerical method, Monte Carlo integration.

6.3 Monte Carlo integration

Monte Carlo integration attempts to approximate a result from an experiment. Monte Carlo methods are usually better for the approximation of higher dimensional integrals than numerical methods [Madras(2002)].

The normalization constant of a Boltzmann distribution for n continuous random variables defined on the same probability space is calculated as:

$$Z = \int_{\mathbb{R}^n} \exp(\frac{-E(t)}{T}) \, dt \tag{15}$$

However, Eq. (15) can without loss of generality also be expressed as follows:

$$E_t\left[\frac{\exp(\frac{-E(t)}{T})}{p(t)}\right] = Z = \int_{\mathbb{R}^n} \frac{\exp(\frac{-E(t)}{T})}{p(t)} p(t) \, dt, \tag{16}$$

where $E(t)$ is an energy function, $p(t)$ can be any probability distribution; $t \in \mathbb{R}^n$, where n is the number of random variables or 3D points and t is the realization of the random variables. $E_t[\cdot]$ is used to represent the expectation value of the function over the random variables.

The Law of large numbers assures the approximation of expected value with a large number of samples and thus Eq. (16) is further reduced to:

$$E_t\left[\frac{\exp(\frac{-E(t)}{T})}{p(t)}\right] = \frac{1}{M} \sum_{j=1}^{M} \frac{\exp(\frac{-E(t_j)}{T})}{p(t_j)} \tag{17}$$
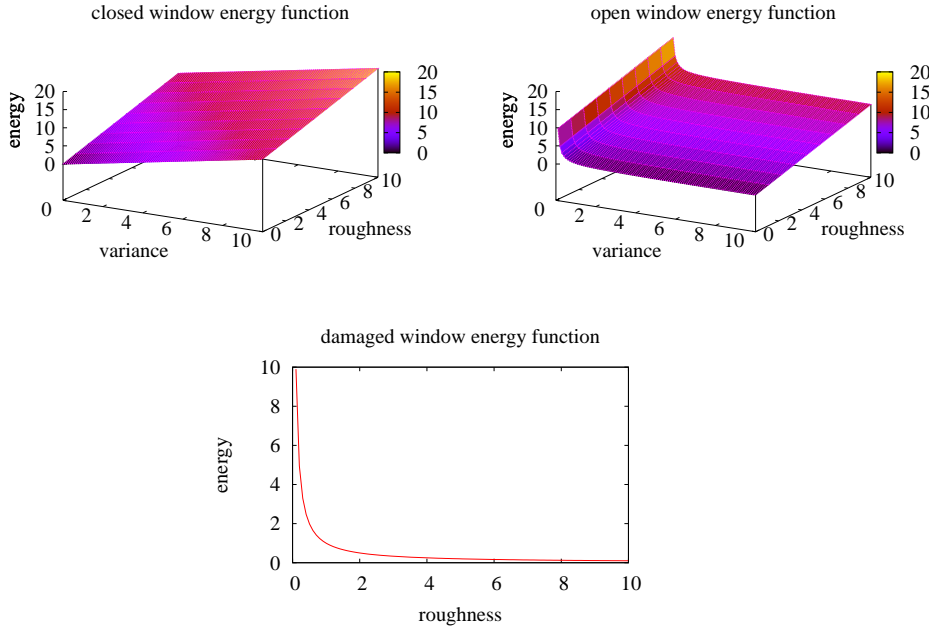
Fig. 8: The energy functions defined on variance and/or smoothness where all the weighting constants are assigned to 1. Top Left: closed window's energy function, Top Right: open window's energy function, Bottom: damaged window's energy function.

Again, $t_j \in \mathbb{R}^n$ is sampled according to $p(t)$ for M number of times, the larger M is the better the approximation will be.

The major practical difficulty in using the Monte Carlo integration is the design of $p(t)$ especially for a function defined in a large space. For example, if we assume $p(t)$ to be uniformly distributed the fraction $\frac{\exp(\frac{-E(t_j)}{T})}{p(t_j)}$ becomes almost infinity since $p(t)$ will be extremely small due to the high dimensionality of the space. Theoretically, this can be solved if we take an infinite amount of samples from $p(t)$. Hence, for practical reasons the issue can only be dealt with a hand designed probability distribution $p(t)$ that tracks the energy function very well– which means, if $\exp(\frac{-E(t_j)}{T})$ is small then $p(t_j)$ is small and vice-versa. Consequently, the fraction $\frac{\exp(\frac{-E(t_j)}{T})}{p(t_j)}$ will be a more pragmatic number that contributes for the estimation significantly. Moreover, we can estimate a more accurate expectation and in effect a better estimate with small number of samples, comparatively, if a well-tracking probability distribution is designed, cf. Fig. 10.

6.4 Designing a probability distribution

The energy function of each label is inherently different, and thus, the design of the probability distribution is also label specific. The goal is to design a probability dis-

tribution that tracks the energy function well, which means $K \cdot p(t) \propto \exp(\frac{-E(t)}{T})$. Therefore, there are two conditions, introduced for convenience, to be satisfied by a probability distribution to be considered as a well designed probability distribution. The first and the foremost is good tracking capability. And secondly, $p(x)$ should NOT be negligibly small regardless of its tracking capability, which is the case for most probability distributions defined in a vast space. An attempt is made to meet the first condition by designing a Markov chain as follows:

$$X_{i+1} = X_i + W \cdot \mathcal{N}(0, \sigma^2) \tag{18}$$
$$X_0 = \mathcal{N}(k, \sigma^2),$$

where i counts the steps starting from 0 to the number of random variables, and X is a random variable taking a temperature value from $\mathbb{R}$, X_0 is the initial state of the process, W is a weight which can be used to control the rate of variation that will be discussed later, and k is a constant that can be tuned accordingly; σ is the standard deviation of the Gaussian distribution. The described discrete time stochastic process Eq. (18) is then ran n times to propose a temperature distribution of a window with n random variables or 3D points. Furthermore, the probability of a temperature distribution instant $t = \{x_n \cdots, x_0\}$ can be calculated as shown in Eq. (19) and Eq. (20). Note that the Markov property is being used to simplify the computation of the joint distribution.

$$p(x_n, \cdots, x_0) = p(x_n, \cdots, x_1 | x_0) p(x_0) \tag{19}$$
$$= p(x_n, \cdots, x_2 | x_1, x_0) p(x_1 | x_0) p(x_0)$$
$$\vdots$$
$$= p(x_n | x_{n-1}) \cdots p(x_2 | x_1) p(x_1 | x_0) p(x_0)$$

And from Eq. (18):

$$p(x_i | x_{i-1}) = \frac{1}{\sigma\sqrt{2\pi}} e^{r^2/2\sigma^2} \tag{20}$$

where r is a number sampled from $\mathcal{N}(0, \sigma^2)$ at the ith step. Thus, Eq. (18) can be used to create a probability distribution that generates sample $t = \{x_n, \cdots, x_0\}$ tracking each label's energy function. However, the free parameters have to be tuned according to the specific energy function; the significant free parameters are W and σ. The second condition for the probability distribution is NOT to be negligibly small. The approach taken towards this problem is *clustering of points* according to their spatial location, and treat each cluster as a random variable, i.e., every point in the cluster will have the same temperature value. This, in effect, will reduce the domain space to the number of clusters from the number of points which means the joint distribution is highly scaled.

The free parameters of Eq. (18) are assigned as $W = 1$ and $\sigma^2 = 0.3$ for *closed window*. Consequently, Eq. (20) and (19) are high valued when there is less variance between consecutive points, i.e., x_{i-1} and x_{i+1} and small when there is high variance and, of course, this by itself encourages smoothness. On the contrary, the variance of *open window's* temperature distribution is much higher than the variance of closed window, *but* the smoothness should be exactly the same. Hence, setting $W = 20.0$ in open window case aims to cause a higher variance between clusters, NOT between points. As a result Eq. (20) and (19) will be high valued

Table 1: Summary of the approximated normalization value and error range.

Label	Number of samples	Normalization constant	Error range
Closed	1000	5.79517	0.0333378
Open	1000	5.26492	0.0231288
Damaged	1000	4.64347	0.0379469

for states with high variance yet smooth temperature distribution. Note that increasing W achieves the proposal of states with higher variance *without causing the probability distribution to shrink* which would happen if we simply increase the variance. Thus, the variance is left as $\sigma^2 = 0.3$.

However, as discussed above, the most distinctive feature of damaged window is the non-smoothness of the temperature distribution, unlike closed or opened window. The design of the probability distribution for the damaged window's energy function is also based on Eq. (18) with minor but basic modifications on the handling of points inside a cluster. In cases of open and closed windows each point in a cluster is assigned exactly the same temperature. But in case of damaged windows the assignment is done as follows:

$$p_j = X_{i-1} + W_2 \cdot \mathcal{U}(0,1) \cdot r, \tag{21}$$

where p_j is a point in a cluster X_i, X_{i-1} is the value of the previous cluster, r is a sample generated from $\mathcal{N}(0, \sigma^2)$ at the ith step, and $\mathcal{U}(0,1)$ is a uniform distribution that is sampled iteratively $\forall p_j \in X_i$. Finally, W_2 is a weighting constant that is used to amplify non-smoothness. As can be seen, each point in a cluster will have different value, unlike the points in a closed or open window. But most importantly, the point's probability is exactly the same as the cluster's, since every value is sampled from $\mathcal{U}(0,1)$ with a probability equal to 1. And this property enables the proposed distribution to represent a very non-smooth temperature, which is expected, without flattening the probability distribution.

Although the hand designed probability distributions are made to track the energy functions well, there is yet another practical issue that is specific to the window labelling problem; *the number of 3D points representing a window vary from window to window.* To address this problem 3D points of the detected window are reduced using an octree sub-sampling. As a result, the number of 3D points on a typical window is assumed to be constant and thus the normalization constant is approximated off-line, which otherwise would have to be computed for each labelling task. Alternatively, for applications where speed is not an issue the normalization constant can be calculated on-line. However, no significant difference has been observed on the final performance except the apparent overhead in the latter case.

7 Experimental results

The test data set is acquired with a highly precise laser scanner, the Riegl VZ-400, and an Optris PI160 thermal camera. The pre-processing, e.g., registration, visualization and mapping of the thermal image is done with *3DTK – The 3D Toolkit* (`http://threedtk.de/`). The window detection is tested on acquired data sets and
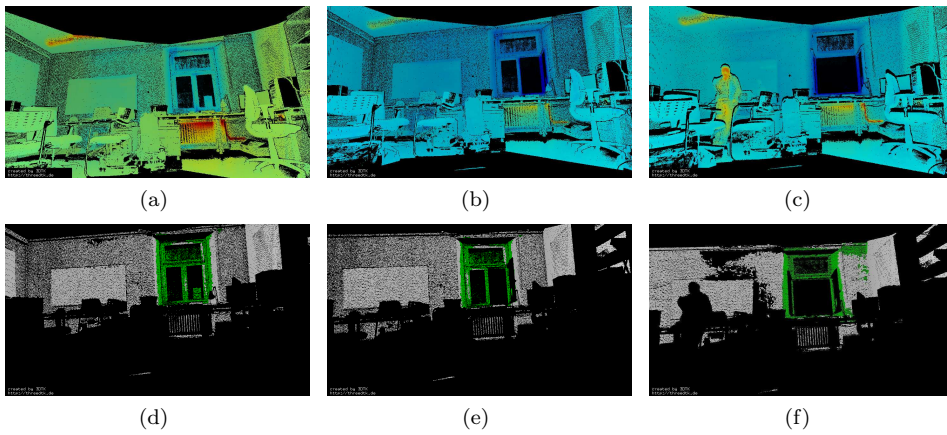
Fig. 9: (a)-(c): Three original thermal 3D point clouds with temperature values. (d)-(f): The detected windows under different circumstances, i.e., (d) closed, (e) semi-open, and (f) fully open are shown in green.

has been proven to be adequate, see Fig. 9. Six registered scans of a laboratory were chosen for initial tests. The room has one window of the type typically existent at Jacobs University (cf. Fig. 2). The data was collected in winter. This season is chosen to reduce the interference of sun on the thermal camera, and to get a clear and distinctive temperature difference between room and outside temperature. Open, semi-open or closed windows are correctly detected. There are, however, free parameters that dictate the shape of the *posterior* probability distribution; the energy function's are parametrized to decrease or increase sensitivity to the features that characterize the specific state. The values of these parameters are hand tuned, for this experiment. However, optimizing these free parameters ensures optimal performance from the specific energy functions. Although, determining the MLE(Maximum likelihood estimator) from the likelihood function– if there is a sufficient training dataset– or using EM(Expectation maximization)– if there are missing variables from the dataset– would suffice for an ideal optimization scenario the difficulty of the normalization constant for analytical treatment has made this road a dead end– keep in mind that the normalization constant is a function of the free parameters. There are, however, other alternatives like *pseudo-likelihood*, where the joint likelihood is approximated by disintegrating it into many spatially independent distributions [Besag(1975)], given there is a huge training dataset to learn from.

The normalization constant for each label is approximated with 1000 samples randomly taken according to the designed probability distribution, see Subsection 6.3. Since the approximation of the mean gets closer to the true mean as $N \to \infty$, where N is the number of samples, the error is estimated with the variance of the following sequence M_h, that gets smaller and smaller as $N \to \infty$, cf. Fig. 10:

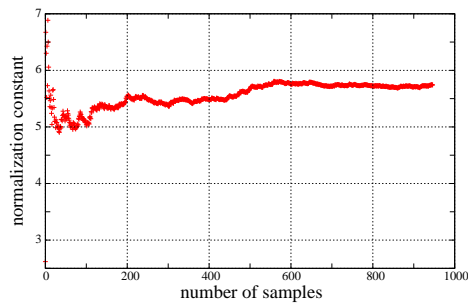$$M_h = \sum_{i=1}^{h} \frac{f_i}{h}, \tag{22}$$

Fig. 10: The convergence of the mean sequence; the approximation of the normalization constant of a closed window with 1000 samples.

Table 2: The value of free parameters for each labels energy function.
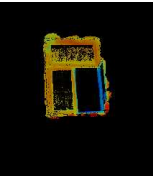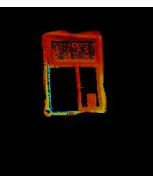
Label	α_1	α_2	α_3	T
Closed	0.05	1	N/A	1
Open	2	1	N/A	1
Damaged	N/A	N/A	0.1	1

where h goes from 1 to N and $f_i = \frac{\exp(-E(t_i)/T)}{p(t_i)}$. As shown in Table 1 the error range of the normalization constants Z for each label is very low. Despite the hand tuning of the free parameters the algorithm performed as expected. The experimentally determined parameters α_1, α_2, α_3, and T are given in Table 2. Exemplary results achieved with these parameters are detailed in Table 3. All examples are correctly labelled by the algorithm. The probabilities for each label and also the probability to choose a wrong label are given. Closed windows are reliably detected. The probability to mislabel an open window is much higher. This is due to the fact that in these cases the non-smoothness increases the probability that the window is damaged. This suggests that the smoothness function is not optimal for window labelling. Nevertheless, also in these cases the correct label was chosen.

8 Conclusions

In this paper, we presented a thermal information analysis for object detection and labelling. The effectiveness of the approach in general for window detection was shown in a small test data set. The method is divided in two parts. First, windows are detected from 3D thermal data. Second, windows are classified into categories based on energy functions of the thermal data. In effect, we presented a reasonable approach for understanding the structure of a room, and we have sequentially shown usage of temperature as a main feature for object detection and furthermore modelling of temperature distribution to infer object related semantics. Although the main aim of this work is to contribute to the efforts of automating energy leak detection and prevention, the approaches can be adopted for object detection and modelling tasks in general, especially in cases where there is a small data set to learn from. One could think of using energy functions to dis-

Table 3: A summary of experimental windows labelling and probability of making an error

| segmented window | Probability of | | | | Final Label |
	closed	opened	damaged	making error	
	0.642699	0.303224	0.054077	0.357301	**Closed**
	0.731546	0.1543682	0.0247715	0.15684535	**Closed**
	0.308042	0.423237	0.1568721	0.576763	**Opened**
	0.271198	0.477311	0.251491	0.522689	**Opened**
	0.347431	0.466858	0.185712	0.533143	**Opened**

tinguish different surface properties based on smoothness or the reflectance value assigned to each point by the laser scanner. In future work we plan to further assess and try to improve the methods presented here. First, an extensive evaluation is necessary using a much larger data set containing windows of different types. Second, this includes a thorough evaluation of the performance under changing temperature distributions. Especially transferring the approach to examine rooms with air condition rather than heating systems is a goal. Second, different energy functions should be evaluated in order to improve the reliability of the labelling. Third, we would like to extend the processing pipeline to label other heat sources and to detect poor insulation in buildings.

References

[Adan and Huber(2011)] Adan A, Huber D (2011) 3d reconstruction of interior wall surfaces under occlusion and clutter. In: 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on, IEEE, pp 275–281

[Besag(1975)] Besag J (1975) Statistical analysis of non-lattice data. The statistician 24:179–195

[Besag and Moran(1975)] Besag JE, Moran PA (1975) On the estimation and testing of spatial interaction in gaussian lattice processes. Biometrika 62(3):555–562

[Besl and McKay(1992)] Besl P, McKay N (1992) A method for Registration of 3–D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2):239 – 256

[Bishop(2006)] Bishop CM (2006) Pattern recognition and machine learning, vol 1. Springer, New York

[Borrmann et al(2008)] Borrmann D, Elseberg J, Lingemann K, Nüchter A, Hertzberg J (2008) Globally consistent 3d mapping with scan matching. Journal Robotics and Autonomous Systems (JRAS) 56(2):130–142

[Borrmann et al(2012)] Borrmann D, Afzal H, Elseberg J, Nüchter A (2012) Mutual Calibration for 3D Thermal Mapping. In: Proceedings of the 10th International IFAC Symposium on Robot Control (SYROCO '12), Dubrovnik, Croatia, vol 10

[Bradski and Kaehler(2008)] Bradski G, Kaehler A (2008) Learning OpenCV, Computer Visoin with OpenCV library, 1st edn. O'Reilly Media

[Clifford(1990)] Clifford P (1990) Markov random fields in statistics. Disorder in physical systems pp 19–32

[Elseberg et al(2011)] Elseberg J, Borrmann D, Nüchter A (2011) Full Wave Analysis in 3D Laser Scans for Vegetation Detection in Urban Environments. In: Proceedings of the XXIII International Symposium on Information, Communication and Automation Technologies (ICAT '11), IEEE Xplore, Sarajevo, Bosnia

[Elseberg et al(2012)] Elseberg J, Borrmann D, Nüchter A (2012) One billion points in the cloud–an octree for efficient processing of 3D laser scans. ISPRS Journal of Photogrammetry and Remote Sensing

[Fischler and Bolles(1981)] Fischler M, Bolles R (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6):381–395

[FLIR Systems AB(2011)] FLIR Systems AB (2011) Thermal imaging guidebook for building and renewable energy applications. In cooperation with the Infrared Training Center (ITC), Sweden

[Fouad and Richter(2012)] Fouad NA, Richter T (2012) Leitfaden Thermografie im Bauwesen – Theorie, Anwendungsgebiete, praktische Umsetzung, 4th edn. Fraunhofer IRB Verlag, Stuttgart

[Ham and Golparvar-Fard(2013a)] Ham Y, Golparvar-Fard M (2013a) An automated vision-based method for rapid 3d energy performance modeling of existing buildings using thermal and digital imagery. Advanced Engineering Informatics

[Ham and Golparvar-Fard(2013b)] Ham Y, Golparvar-Fard M (2013b) EPAR: Energy Performance Augmented Reality Models for Identification of Building Energy Performance Deviations between Actual Measurements and Simulation Results. Energy and Buildings

[Holz et al(2011)] Holz D, Holzer S, Rusu RB, Behnke S (2011) Real-time plane segmentation using rgb-d cameras. In: Proceedings of the 15th RoboCup International Symposium, Istanbul, Turkey

[Hornung et al(2013)] Hornung A, Wurm KM, Bennewitz M, Stachniss C, Burgard W (2013) OctoMap: An efficient probabilistic 3D mapping framework based on octrees. Autonomous Robots DOI 10.1007/s10514-012-9321-0, URL http://octomap.github.com, software available at http://octomap.github.com

[Jordan(2002)] Jordan A (2002) On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. Advances in neural information processing systems 14:841

[Laguela et al(2014)] Laguela S, Daz-Vilarino L, Arias JAP (2014) Non-destructive approach for the generation and thermal characterization of an as-built bim. Construction and Building Materials 51:55–61

[Luhmann et al(2010)] Luhmann T, Piechel J, Ohm J, Roelfs T (2010) Geometric calibration of thermographic cameras. In: International Archives of Photogrammetry, Remote Sensing and Spatial Information, Newcastle upon Tyne, UK, vol 38 part 5, commission V Symposium

[Madras(2002)] Madras N (2002) Lectures on Monte Carlo Methods. Fields Institute for Research in Mathematical Sciences Toronto: Fields Institute monographs, American Mathematical Society

[Pangercic et al(2012)] Pangercic D, Tenorth M, Pitzer B, Beetz M (2012) Semantic object maps for robotic housework - representation, acquisition and use. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '12), Vilamoura, Portugal

[Pedersen and Hellevang(2008)] Pedersen C, Hellevang K (2008) Insulation and air infiltration levels. Circular AE (NDSU Extension Service) 1373

[Schnabel et al(2008)] Schnabel R, Wessel R, Wahl R, Klein R (2008) Shape recognition in 3d point-clouds. In: The 16-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2008

[Surmann et al(2003)] Surmann H, Nüchter A, Hertzberg J (2003) An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. Journal Robotics and Autonomous Systems (JRAS) 45(3–4):181–198

[Vosselman and (Eds.)(2010)] Vosselman G, (Eds) HGM (2010) Airborne and Terrestrial Laser Scanning. Whittles Publishing, Scotland

[Xiong and Huber(2010)] Xiong X, Huber D (2010) Using context to create semantic 3d models of indoor environments. In: British Machine Vision Conference, Aberystwyth, pp 45–1