

UNIVERSITY COLLEGE LONDON

ENGD GROUP PROJECT

A New Approach to Thermal Imaging Visualisation

Thermal Imaging in 3D

Author:

Jessica WARDLAW
Maciej GRYKA
Fabian WANNER

Supervisor:

Gabriel BROSTOW
Jan KAUTZ

July 30, 2010

Abstract

Heating and cooling efficiency is very important in today's world because heating is expensive due to the high oil and energy costs. Furthermore, global warming has to be stopped as quickly as possible.

Some of the main causes of energy wastage are small gaps in the outside of buildings, bad insulation or inefficient windows. All of these can be detected with the help of thermal imaging cameras. Currently there exist a number of companies that specialise in such services, however, we are not aware of any that perform 3D reconstruction of obtained models.

This project analyzes the effect of a new representation for thermal images on the ability of users to find and locate heat leakages on a building. This new visualization is a rough 3D model of a building. To achieve this, we implemented an algorithm, which creates a simple geometric representation of a scene, which it then textures with visual as well as thermal images to allow users to switch between two views. It also allows switching between images taken at different times.

This report will present a user evaluation in which we compared users' ability to detect heat losses using normal thermal images with our 3D model and also obtained feedback on the Graphical User Interface (GUI) that we had designed. Non-expert users were able to interpret otherwise complex thermal data and to start explaining the patterns they see because the model facilitated a global overview of the building and put each thermal image into context. The model allowed for a much richer and more intelligent interpretation of the thermal data, with users becoming more confident of their assessments.

Acknowledgements

We would like to thank Gabriel Brostow and Jan Kautz for their support and advice during the project, and also for their enthusiasm and interest in our progress.

Thanks to Tim Weyrich for supporting our search for a thermal camera successfully.

Our gratitude goes to Dejan Mumovic for lending us the thermal camera of the Energy Institute, for the introduction to building science and explaining the camera and the problems of thermal imaging.

Contents

1	Introduction	4
2	Thermal Imaging	5
2.1	Building Science	5
2.1.1	Emissivity	5
2.1.2	Atmospheric particles	6
2.1.3	Ambient temperature and wind effects	6
2.1.4	Image angle and distance effects	6
2.2	Using IR for Building Inspections	6
3	Image Acquisition	7
3.1	Cameras	7
3.2	Acquisition Challenges	7
3.3	Physical Setup	8
4	Reconstruction	9
4.1	Point Cloud	9
4.2	Triangulation	11
4.3	Texturing	11
5	User Evaluation	14
5.1	Aim	14
5.2	Method	14
5.3	Results and Analysis	16
5.4	Discussion	19
5.5	Conclusions	19
6	Extension - Plane Fitting	20
6.1	Plane Detection	20
6.2	Clustering	22
6.3	Shape Approximation	23
6.3.1	Quad Trees	24
6.3.2	Image Morphing	25
6.3.3	Border Detection	26

6.4	Polygon Fitting	27
7	Conclusions	28

Chapter 1

Introduction

This project analyzes the effect of a new representation for thermal images on the ability of users to find and locate heat leakages on a building. This new visualization is a rough 3D model of a building. Our hypothesis is that the 3D model will make it easier for users to get a global overview of the building and immediately detect the exact location of the problem because they are able to look at the different parts of the building from all directions.

To test this, we implemented an algorithm, which creates a simple 3D model of a scene, which it then textures with the visual as well as the thermal images to allow users to switch between two views. It also allows switching between thermal images taken at different points of time.

This report will first discuss the context of the project from the building science perspective and the factors that must be taken into consideration in the interpretation of thermal infrared images. The acquisition of the thermal data for this project will then be discussed, followed by the construction of the 3D model. Further, the report will present a user evaluation in which we compared the heat loss detection using normal thermal images and the 3D model. The results will be described in the corresponding chapter. Finally we will introduce an extension to our method that fits planes to the point cloud rather than triangulating it. This approach should work well for buildings and create simpler and cleaner models.

Chapter 2

Thermal Imaging

2.1 Building Science

Infrared light falls between the visible and microwave parts of the electromagnetic spectrum with wavelengths of 2-15 μm , so is not visible to the human eye. Near IR waves (0.7-25 μm) are closest to visible light; far IR waves (25-1000 μm) are closer to the microwave region.

Any object with a temperature higher than absolute zero (-273.15°C or 0 K) radiate energy as electromagnetic waves, which travel at the speed of light. For temperatures commonly encountered in building physics, most emissive power falls within the IR part of the electromagnetic spectrum; the higher the temperature of an object the more IR radiation it emits.

Thermal imaging devices make an image of the thermal patterns and measure the emissive power of surfaces in an area at various temperature ranges. They use a lens to focus the emitted IR radiation onto a detector and the electrical response signal is converted into a visual display (digital picture) in which the different colours correspond to various temperature levels of the surface on which it is focused. These images can be analysed to identify the potential problem areas. If the surface emissivity is known, within the spectral range of the IR detector, temperatures can be calculated (Balaras and Argiriou [4]). The accuracy of measurements depends on a variety of factors:

2.1.1 Emissivity

Emissivity is a measure of how efficiently a surface emits energy compared to a surface at the same temperature that is a perfect emitter (blackbody) with an emissivity equal to one. Accurate readings are not possible on surfaces with an emissivity of about 0.5 or lower e.g. unpainted, clean and shiny metal mechanical and electrical components. High emissivity (around 0.9) targets may be attached to shiny metal surfaces to be inspected e.g. paper stickers or paint.

2.1.2 Atmospheric particles

Large size particles in the atmosphere (i.e. water vapour, carbon dioxide gas molecules and ozone) attenuate radiation; dust particles and water droplets produce an additional scattering effect. Since the absorption varies with the thickness of the gas traversed by the radiation, the effect is not a measurable constant and cannot be compensated.

2.1.3 Ambient temperature and wind effects

Ambient air temperature influences the temperature of the equipment and its performance; IR equipment compensates for this variation internally. It may also affect the amount of IR radiation emitted from the surface. High winds will enhance heat transfer from the surface and higher convective heat losses can reduce the surface temperature. Measurements should thus not be taken outdoors in windy conditions, particularly if speeds exceed 5m/s. High winds can, however, make it easier to identify air infiltration problems around openings interior windows and doors.

2.1.4 Image angle and distance effects

Distance effects and the angle of vision also affect the interpretation of thermal images. The resolution of the image decreases with distance, with each image pixel representing a larger surface area. The radiation emitted from that area is averaged and detail is lost. Viewing an object at an acute angle presents less information than one taken at right angles.

2.2 Using IR for Building Inspections

IR inspections can be used to detect heat losses from buildings, missing or damaged thermal insulation in walls and roofs, thermal bridges, air leakage and moisture sources. Interventions can then be well targeted with a reduced repair cost, while saving heating and cooling costs. Images are used as a proof of a problem or support other data collected during an audit to substantiate proposals. Accurate and detailed information of where a problem occurs can reveal the cause of the problem, thus avoiding extensive and expensive renovation work through low-cost local repairs.

IR measurements should be taken at night or during a cloudy day, with low wind speeds to minimise convective heat losses. Alternatively, measurements can be taken on interior walls to minimise the impact of adverse outdoor conditions. It should be noted that the impact of absorbed solar radiation continues for a few hours.

Chapter 3

Image Acquisition

This chapter describes the cameras that were used to acquire the visual and thermal images with emphasis on the characteristics of the thermal camera and the challenges we faced during the acquisition process.

3.1 Cameras

Two cameras were used: Flir ThermaCam S65 for thermal and Canon IXUS 120IS for capturing visual images. The reason we decided to capture both simultaneously are twofold: firstly the system was designed to make it easy for users to examine buildings and this task is aided by the ability to switch between different views (see [section 5.3](#)). Secondly reconstructing a 3D point cloud from thermal images alone proved difficult for reasons explained in [section 3.2](#). The visual images were captured with a simple point-and-shoot style device to show that in this respect no special constraints need to be satisfied. See [Figure 5.2](#) for an example of matching visual and thermal images.

3.2 Acquisition Challenges

Images captured with thermal cameras have the following characteristics that make it difficult to perform 3D point cloud reconstruction from them alone:

- Long focal length means that it is difficult to capture a large part of a building facade, especially in the urban environment. This has a negative impact on users' ability to interpret the images, but also makes it difficult for SFM (Structure From Motion) algorithm [20] to perform reconstruction as fewer SIFT [16] correspondences can be found. We overcame this problem by taking additional visual images (without thermal counterparts) with a wider field of view. The point cloud was then reconstructed from all available visual images allowing us to obtain camera positions for thermal images from corresponding visual images.



Figure 3.1: Acquisition setup - two cameras were fixed on a tripod so their relative position stayed constant.

- The particular model of the camera we used left an unremovable company logo in every photograph taken making it impossible to perform structure from motion reconstruction as the logo contained many very strong features. Cropping the images to remove it would have meant losing even more field of view which would attenuate the problem mentioned above. This issue resurfaces in the texturing stage, where it leaves artefacts on the textured model - see [Figure 5.3](#).

3.3 Physical Setup

3D reconstruction performed on visual images alone required a way of relating them to the corresponding thermal images in order to texture-map the model. We achieved this by fixing both cameras' positions with respect to each other and taking images simultaneously. This results in a data set consisting of a number of visual images with their thermal counterparts taken at the same time and from (approximately) the same position. Because in our case the distance between the cameras was small relative to the objects we were photographing we did not take any pixel-offset into account. Furthermore the alignment does not need to be pixel-perfect in order for users to easily identify areas of potential problems. The setup used is shown in [Figure 3.1](#). The images were captured at two different times (5 pm and 7 pm) to allow users to investigate how the heat properties of the building change with time.

Chapter 4

Reconstruction

This chapter presents the pipeline we designed to create a textured 3D model. In summary, having the 2D images as an input we ran a Structure From Motion (SFM) algorithm to obtain a 3D point cloud. Next, we reconstructed a 3D model by triangulating the points. Finally textures were automatically applied to polygons based on camera positions estimated by SFM. The overview is presented in [Figure 4.1](#). The following sections explain each step in more detail.

4.1 Point Cloud

The first step to achieve the reconstruction is to obtain a dense point cloud. Here we used a very common approach, as this is not the main objective of the project and thus decided to use Bundler [19, 20], Clustering Views for Multi-view Stereo [13, 14] and Patch-Based Multi-View Stereo [15]. These three systems have been combined by James Tompkin [21] in a script, so that the information is correctly passed from one program to the other.

Bundler [19, 20] uses SIFT [16] and Bundle Adjustment [22] to create a sparse point cloud by looking for features in the images, matching them and then ensuring that the 3D position of the point corresponds in all the images. This is important as the matching algorithm in SIFT [16] returns many false matches.

An example can be found in [Figure 4.2](#). The obtained result provides very good information about the building of interest, however, it also contains points from surrounding objects and other buildings. This, being an urban environment, is practically impossible to avoid and so in our current implementation we removed this “noise” manually. An extension to our method described in [chapter 6](#) would overcome this problem by automatically finding major planes rather than fitting a closed mesh.

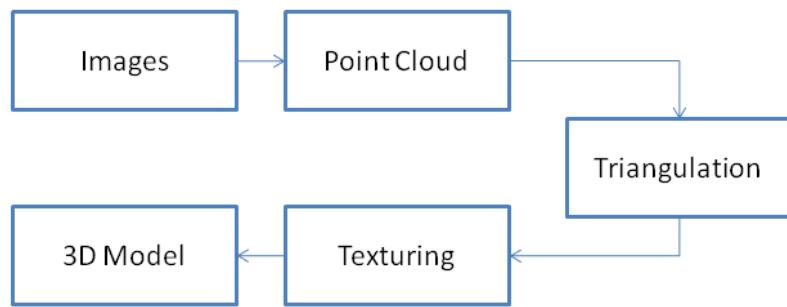


Figure 4.1: Overview of our pipeline.



Figure 4.2: Point Cloud of a Building before (left) and after removing unnecessary points (right).

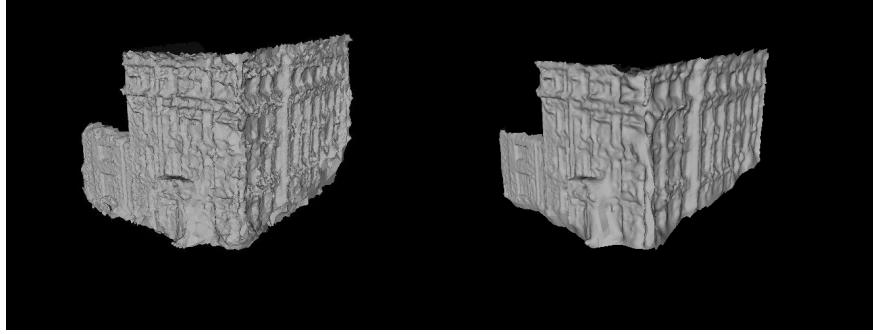


Figure 4.3: Triangular meshes reconstructed from the point cloud after clean-up. The reconstruction can be done either with CGAL [2] (left) or MeshLab [1] (right).

4.2 Triangulation

The next step was to reconstruct a triangular mesh from the dense point cloud. We experimented with both using CGAL’s [2] function that takes as an input a point cloud with oriented normal vectors at each point and outputs a reconstructed surface and equivalent MeshLab [1] functionality. CGAL’s main advantage was that it could be easily called from code, while MeshLab had to be interacted with manually. However, since the manual clean-up after the triangulation had to be performed anyway, this was not introducing any additional limitations. Moreover MeshLab produced the models much quicker for the same number of points (the order of seconds rather than hours as was the case with CGAL). This is likely due to sub-optimal parameters that were passed to CGAL and the times might have been significantly improved given the right values - in fact CGAL’s manual [3] suggests that for 100k points the process should take around 2 minutes. We, however, were unable to get this performance.

The manual clean-up was performed after this stage as typically we were only able to model a few faces of the building (it might be very difficult to capture a whole 360°view of a building in the city) and fitting a closed mesh produced spurious polygons. The result of the whole procedure is illustrated in Figure 4.3.

4.3 Texturing

The last part of the reconstruction pipeline was to texture the model. In our implementation each triangle is textured separately using image from one of the camera positions provided by Bundler [20]. First, based on the normal and the position of a given triangle a camera has to be chosen. Each camera is given a score based on its viewing direction and position. The viewing direction is scored as the inner product between the surface normal and camera orientation - value

close to 0 means that the vectors are almost perpendicular, while value close to 1 means that they are almost parallel, which is essential in order to obtain the right texture. The position on the other hand is scored as the distance of the camera from the line coming out of a centroid of the triangle in the normal direction.

After the best camera for a given triangle was found, we used its intrinsic and extrinsic parameters to project 3D points defining the polygon into a camera image. If the projected point fell outside of the camera image, another camera was sought. The resulting 2D coordinates acted as our UV texture coordinates for this particular triangle.

It is important to note that the above solution only works when there are cameras looking at every wall that is of interest. In our case Bundler failed to register some of the cameras. This is most likely due to not being able to find enough correspondences in the images, which was in turn due to small overlap between images. Unfortunately that meant that we did not have any valid camera (i.e. one with both visual and thermal images) directed at one of the walls of the example building. This can be seen in [Figure 4.4](#) (top row) - the cameras with long focal lengths are the ones with both visual and thermal data, while cameras with short focal lengths only have visual data. As can be seen no thermal data was obtained for one side of the building. As a result this wall was textured improperly ([Figure 4.4](#)).

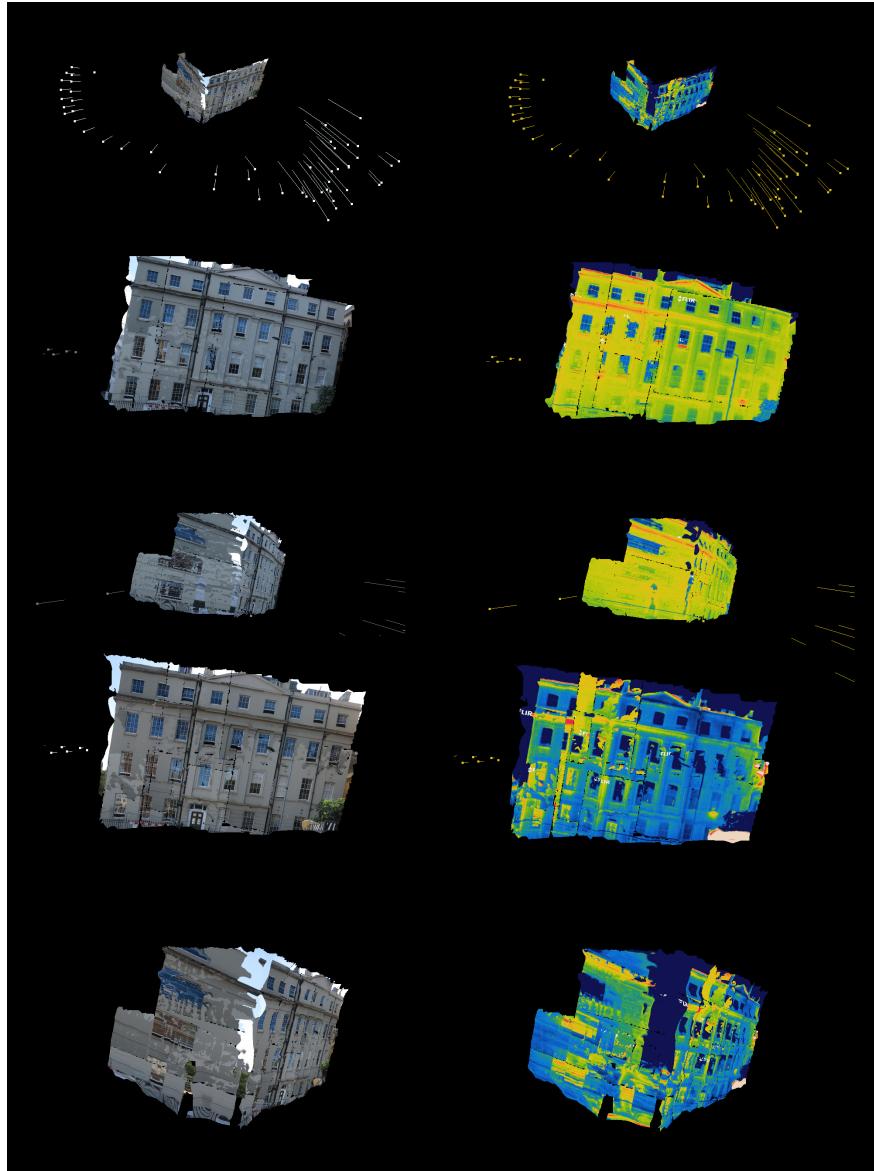


Figure 4.4: Final output of our system - textured 3D model of the building with both visual (left) and thermal (right) information at two different times: row 2 and 3 present data from 7pm, while 4 and 5 from 5 pm.

Chapter 5

User Evaluation

5.1 Aim

One of the objectives of our project was to provide an intuitive interface for the 3D Model. The aim of the user evaluation was thus to review the interface design and investigate if our 3D model helped people with limited knowledge of Building Science to interpret thermal data so that the system can be used by both novice and expert users.

5.2 Method

In total 13 people took part in the user evaluation. One participant was an expert in Building Science. The remaining 12 participants were students who did not have such expertise, although one had some background in architecture (but was unacquainted with thermal data) and one had experience of Human-Computer Interaction, who was recruited for expert feedback on the interface itself. Evaluations were carried out with versions of our system that contained manually texture-mapped data.

Participants were asked the following four questions with the objective to get direct feedback on the interface to direct future development and to also find out if the 3D model facilitated a quicker and more detailed understanding of the thermal data:

1. Please identify where the building is losing heat and try to explain any differences in temperature that you see.
2. How does this other representation compare? Is there anything missing that was there before or does it show you something new that you could not see before?
3. Which do you prefer, 2D or 3D, and why?

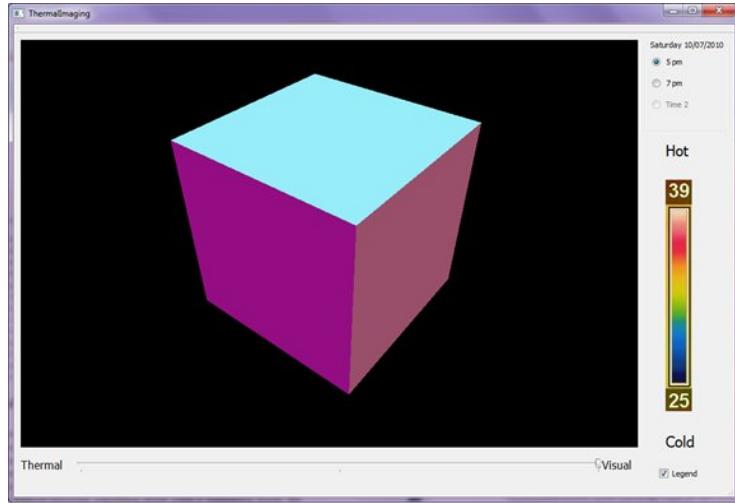


Figure 5.1: The “Cube“ application used for demonstration

4. Which data should the 3D model display when it is first opened: thermal, visual or a combination of both?

For the first two questions, participants were encouraged to "think aloud" in their answers. This is a common method used in usability evaluations and was chosen because it would give insight into the participants' development of thought [8]. Verbalizations, which take place simultaneously with cognition, are mostly free of interpretation and as such reflect the cognitive processes that generate behavior or action [23]. In this way, when subjects "think aloud" one gains insight into their short term memory [9]. The final two questions were asked after the experiment. Since users can be incoherent when "thinking aloud" [9] due to the cognitive load of problem solving and speaking concurrently, short post-experiment interviews often provide more articulate responses and make think-aloud data easier to understand and interpret [6], [12].

Before each evaluation we demonstrated to participants how to interact with the system. For the demonstration a cube was used instead of the 3D building, so that participants would not see the thermal data prematurely (Figure 5.1). Half of the 12 student participants were shown the 2D images first and half were shown the 3D model first to mitigate for any influence that the order in which they were shown the 2D images and 3D model may have had on responses. Throughout the experiment we alternated the data first visible when the program was opened to control for any biased answers to question 4. The thermal data had not been integrated in time for the first six student evaluations so they were shown pseudo-thermal data; this is worth noting as it may have affected their interpretation of the thermal data (but not the interface itself).

5.3 Results and Analysis

In general, participants were able to pick out more detail on the 2D images compared to the 3D model (e.g. trees, railings, pipes, roof artefacts, balcony, columns, door knocker) and start to theorise possible explanations for the thermal patterns they saw. This, however, can be largely attributed to the poor resolution of the images when rendered in 3D, which has been improved in the latest implementation of the system by using full, rather than under-sampled textures ([section 4.3](#)). The expert evaluator suggested that, if the image resolution is improved, the system could be used to detect thermal bridges. This is a phenomenon in which heat escapes through gaps in the insulation of buildings (particularly building corners) and account for 30-35% of heat losses from buildings built in the 1960s and 1970s.

Despite seeing more detail in the 2D images, participants often found it difficult to identify the features they could see in the thermal images without the visual data to provide context e.g. pavement, road barriers, lamp post, boarded-up windows; they often had to guess what they were looking at and made errors. For example, a couple of participants thought a lamp post was in fact a drain pipe ([Figure 5.2](#)), and identified road barriers as the building basement. By integrating the images taken in the visual part of the EM spectrum at the same time as the thermal data, such interpretation errors were greatly reduced. The lack of detail and simplicity of our 3D model somewhat prohibited this; interestingly, one participant said that some of the 2D images appeared 3D due to the angle of the camera.

When browsing through the 2D images participants were required to use the arrow keys on the computer keyboard to scroll through them. The transition between images was thus very cumbersome and hindered interpretation. Participants gave positive feedback to the smooth interactions afforded by the 3D model, which made them feel more "in control".

Participants found it very difficult to visualise the building layout from the 2D images and were unable to attribute the photos to particular sides of the building. In contrast, the 3D model revealed the building layout to the extent that participants were able to start identifying heating pipes, windows, doors and steps. Furthermore, participants began to theorise that temperature contrasts were caused by the angle and direction of the sun and therefore whether they were looking North, South, East or West ([Figure 5.3](#)). This was aided by the temporal aspect of the 3D model; participants talked about the sunset and the participant who had studied architecture talked about "thermal lag", the time it takes building materials to absorb and lose heat. By knowing the building layout, participants were able to interpret the thermal data much more quickly using the 3D model. One participant, who saw the 2D images first, took almost three minutes to start talking about the thermal data.

In addition, participants provided feedback on the design of the interface. Many participants found it difficult to remember which mouse buttons they should use for which action, despite the tutorial before the experiment. One participant suggested a help section would be beneficial, to remind users which



Figure 5.2: Identification mistakes. This lamp post was mistaken for a drain pipe.

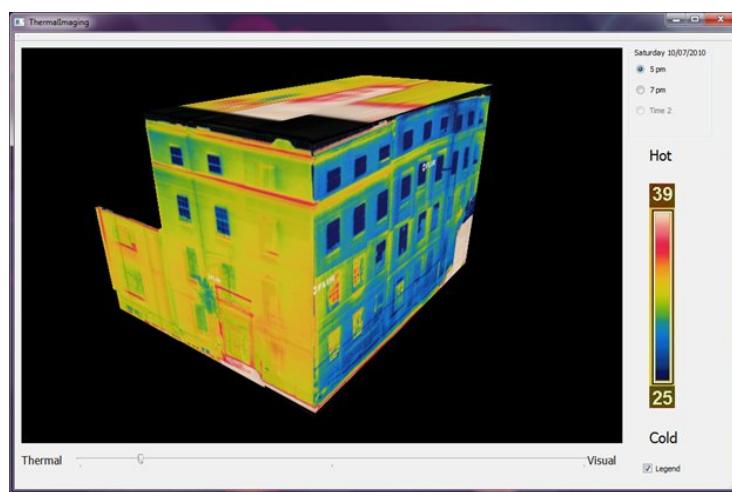


Figure 5.3: Building facade. This image demonstrates the effect of sun angle on building temperature.

mouse button performed which action. Participants also found it difficult to hold down the scroll button for panning; they suggested that it would be better to use one mouse button for rotation and the other for panning, which was implemented in the latest revision of our system. One participant found panning so difficult that they suggested a reset button in case they inadvertently "dragged" the building off screen.

The feedback on the slider bar, which blends the visual and thermal data, was positive. Interestingly, despite the demonstration and opportunity to practice using the cube, some participants still did not use the slider bar until prompted. This may be an indication that users found this particular action hidden within the interface. One participant said that having the visual and thermal data side-by-side might be more intuitive so that users are more aware that there are two datasets available. The Building Science expert suggested that we use temperature for the transition, so that the colder areas fade to thermal first and the user sees only the warmest parts of the visual image; the effect of the slider bar wasn't immediately obvious to him. This is a possible adaptation for future development.

Only a couple of participants noticed that the legend changed when they clicked between the 5pm and 7pm data, indicating that many users would miss it and assumed that the legend was the same for both times. Inconsistent legends would lead to erroneous interpretations of the data and prohibit comparison of data captured at different times. The legend can be fixed when acquiring the data, but some of the detail in the temperature measurements would be lost as a result.

For more accurate and detailed interpretations, the expert suggested that contextual information such as the weather or a compass (so that user knows which direction they are looking in) would be very helpful. As previously stated, there are many factors that influence the quality of thermal data, such as air temperature and quality, and this is important information that users need in order to understand the reliability of the data, particularly if they are experts.

Most negative feedback on the 3D model related to the resolution of the thermal data and the accuracy with which it is aligned to the visual data. Thermal data for the roof is also necessary since most of building heat loss occurs through the roof. These would be the priorities in any future development work.

Finally, feedback was sought on which data the user would prefer to see when they first open the program; is it better that users first see visual data, thermal data or a mixture of both? Some participants said visual would be most useful because it helps the user to understand the building structure and to "acclimatise to the environment". Others said that starting the system with the thermal data would result in more objective interpretations of data because they would have no preconceptions about the building structure; furthermore, the system has been designed for interrogation of thermal data so it is better to start there and move to the visual data. Other participants said they would prefer the system to start with a blend of thermal and visual data because it prompted them to look at both datasets and they felt it made it easier to move to either the thermal or visual data, as they wished.

5.4 Discussion

A criticism of the methodology applied here might be the validity of comparing 2D thermal to 3D thermal images plus visual; it could give an unfair advantage to the visual data. It should be noted though that when users scrolled through the thermal images there were visual images in the same folder location; some participants looked at these visual images to identify what they were looking at. Another experiment might only compare the thermal data in both dimensions. For the purposes of this project though it was desirable to evaluate the system as a whole.

The expert in Building Science reported that the thermal data in the system was misleading because it was captured on a hot day; infrared cameras tend to measure surface temperature when solar radiation is intense. The thermal mass of buildings (i.e. their capacity to store heat) will also affect images taken on hot days. The temperature of the sky and the huge reflection from the windows may have also affected the temperature range on some images, increasing the difficulty of interpretation. The best time of day to capture thermal data is early in the morning or later at night when it is much cooler.

Fundamentally, we discovered that the system would need to quantify the heat leakages for it to be useful for experts, which is not possible without integrating thermal models or measurements from inside the building. Until this is done, experts would not benefit from using the system as they are able to identify where heat will be lost from a building from a visual assessment. It would also be preferable to conduct the study with more participants; within the time frame of the project it was not possible to test the system with more experts and more students of Civil Engineering and Architects. This would be the next stage of any evaluation.

Another issue that came up during the evaluations was that some people find it difficult to conceptualise and interact with 3D objects on a flat computer screen. For example, holding down a mouse button and drawing a circle to rotate a 3D object is a particularly abstract concept for these people. More research needs to be done on this phenomenon and how such users might be assisted when interacting with 3D objects, particularly since interaction is no longer restricted to direct manipulation.

5.5 Conclusions

The evaluation has showed that the 3D model we developed helps novices to interpret otherwise complex thermal data and to start explaining the patterns they see. It allows for a much richer and more intelligent interpretation of the thermal data, with users becoming more confident of their assessments. Whilst we concede that there are improvements to be made in the interface, users were generally more engaged with the 3D model than they were with the 2D images. Engagement is especially important for such cognitive tasks as interpretation of thermal data ([5], [7]).

Chapter 6

Extension - Plane Fitting

This chapter presents an extension to the above solution which allows us to fit planes to the 3D model. Doing this eliminates the need to manually clean-up the point cloud and later the 3D model as long as the dominant planes belong to the object of interest. This is a reasonable assumption because the points belonging to background are unlikely to form major planes (they are only seen in few images).

To achieve this we developed a pipeline that is a slight modification to the one presented in [chapter 4](#). This is illustrated in [Figure 6.1](#) followed by an explanation of all the parts in detail.

6.1 Plane Detection

The next step in the extended pipeline was to detect points that resemble planes in the scene. The technique we chose for this was RANSAC [11]. It is a relatively slow algorithm, but it gives good results even if the number of outliers is very

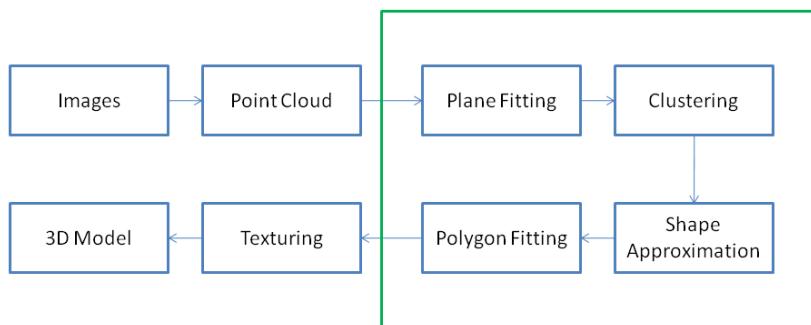


Figure 6.1: Extended reconstruction pipeline.

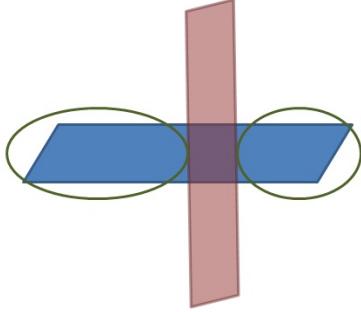


Figure 6.2: Problem with intersecting planes.

high if the number of iterations is appropriate. Myatt et al. [17] showed that RANSAC works well in low dimensional spaces.

RANSAC [11] can be illustrated by the following pseudocode:

```

maxInliers ← 0
bestPoints ← []
for EveryIteration
    randomPoints      comment: three random points
    planeEq          comment: find plane equation
    inliers ← 0
    for EveryPoint do
        distance      comment: distance plane to point
        if distance < inlierDistance
            inliers ← inliers + 1
        fi
    od
    if inliers > maxInliers
        maxInliers ← inliers
        bestPoints ← randomPoints
    fi
od

```

In the actual code, we also kept track of which points were considered as inliers, so that we knew which points to work with in the following steps.

One additional problem we encountered was that the same plane was found in many iterations. However, we could not just ignore all the points that have been used already, as this would cause problems in the case of a plane intersection. This is illustrated by Figure 6.2 which shows two planes, the red one and the blue one. Let us assume in the first iteration RANSAC found the red plane. If we removed all the points on that plane from future processing, our blue plane could be split into pieces.

To avoid this, we treated points that have not been located to be on a plane

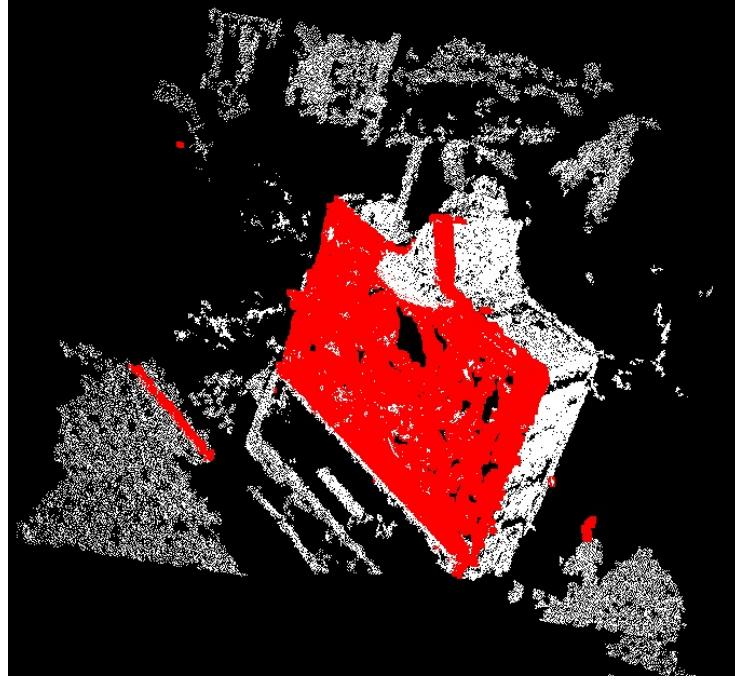


Figure 6.3: Plane problems because of infinite planes.

and those that have been added to a plane differently. When choosing random points and counting inliers, we do not consider those points that have been allocated to a plane before. However, when adding points to a plane (counting inliers), we do consider all points.

This means, we do not get the same plane on every iteration, but we also avoid gaps in the planes.

Here it is important to mention that after this step we rotate all the points on this plane to the XZ plane and flatten them, so that they have a y-value of 0. All the steps after this happen in 2D space, so aligning the points along the coordinate axes and thus being able to ignore one coordinate will simplify the following steps.

6.2 Clustering

After the previous step we had an infinite plane in the scene representing a face of the object. For example in Figure 6.3 it can be seen that the dominant plane shown in red (the side of a box) also includes parts of the floor and other objects. In order to create a realistic 3D model we needed to find its limits, which was done by clustering of the contributing points.

To overcome this we had several options:

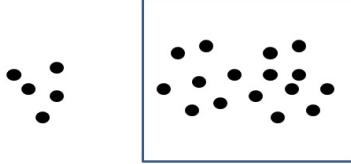


Figure 6.4: Clustering the points on the plane and finding the biggest cluster.

1. Create one big plane combining the clusters
2. Create a plane for each cluster
3. Create a plane for the biggest cluster

Option one is not suitable, as it does not represent the true plane set up and it also might cause a plane in the front to cover the entire field of view. Option two, would be a possibility, as the errors produced by this approach are not very large. However, it might create many unnecessary planes, which can limit the view at the scene from some perspectives.

Method three seems most promising and thus we decided to implement this. The advantage of it is, that even if option two would have been the best, another plane containing all the points missed out here should be found in a later RANSAC iteration. [Figure 6.4](#) shows that idea diagrammatically.

This type of clustering was implemented in two steps:

1. Cluster all the points on the plane
2. Take all the points on the biggest cluster and remove all the others from the list of points for that plane

Part 1 starts by placing the first point into cluster one and all the points that are below some threshold distance away are placed in the same cluster. Then we look for close neighbors for all the points that have just been added to the list and continue doing this until no more points are found that are close to the newly found ones.

Then we take the first point in the list that was not clustered yet and place that in cluster two. The same approach is taken here. This is done as many times as required to place every point in a cluster.

Part 2 takes the biggest of the clusters found in Part 1 and removes all the other points from the list of points for that plane. This is to make sure that these points can later be used for another plane.

6.3 Shape Approximation

From the cluster we need to find a border for the plane. The simplest method would be to look for the minimum and maximum x and z value, but it does not work for triangular planes or even rectangular planes that are rotated.

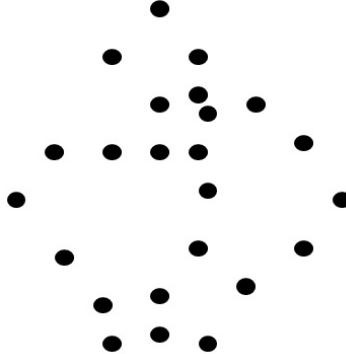


Figure 6.5: Sample point cloud after clustering.

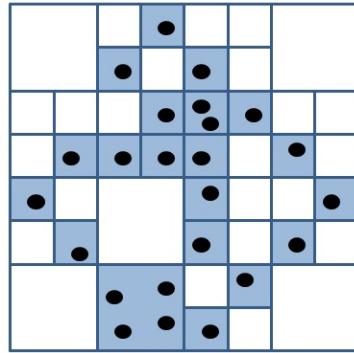


Figure 6.6: Quadtree decomposition.

We are going to use the point cloud in [Figure 6.5](#), as an example for the following few steps. This point cloud is assumed to have been clustered already, so all parts on the plane, which are too far away from the main cluster have been excluded. It has approximately the shape of a rotated square.

6.3.1 Quad Trees

As already mentioned, the minimum and maximum of x and z values does not give any good results in most situations.

Here we took a different approach and first applied quad tree decomposition [10] on the 2D bounding box of the point cluster. Each leaf of the quad tree will be marked depending on whether there are any points in it or not. This will return a rough structure of the shape of the plane, as seen in [Figure 6.6](#). The blue fields mark those, which contain points and the white fields do not contain any.

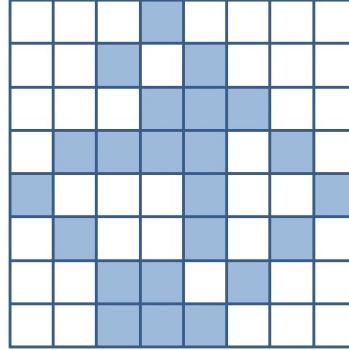


Figure 6.7: Grid obtained from Quadtree

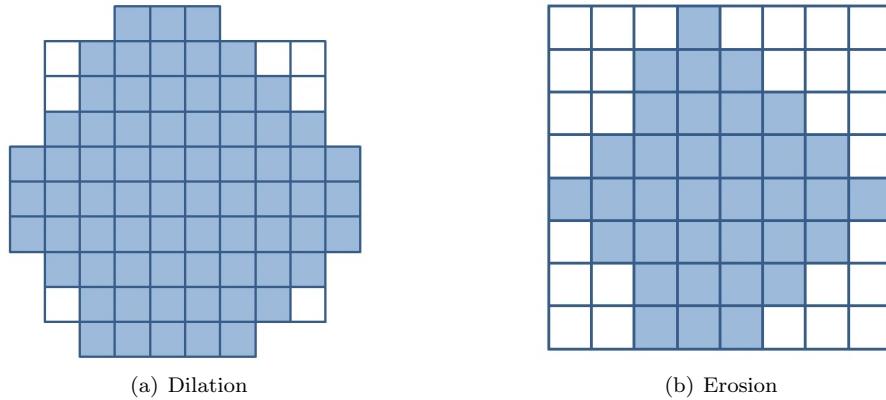


Figure 6.8: Dilation and Erosion

6.3.2 Image Morphing

It can be seen that this resembles more or less the shape of the point cloud, but has several problems. There are holes in the middle and in the border and there can be noise on the outside. To overcome these two problems, we first transfer the quadtree into a grid (or binary image), as shown in [Figure 6.7](#).

This will allow us to perform several image morphing operations. Dilation and erosion are of interest in this particular case. Dilation can be used to fill in holes in the binary image and erosion is useful for removing noise on the edges of the border. These steps always have to be carried out consecutively, so that the shape is not altered in any unexpected way, i.e. we can dilate the image n times and then erode it n times to remove holes in the binary image or we can erode n times and dilate the same number of times to remove noise on the edge of the structure.

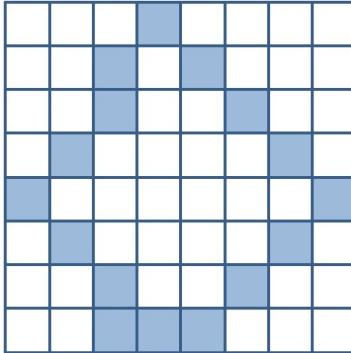


Figure 6.9: Border of Approximate Shape

Figures 6.8(a) and 6.8(b) show a single step of dilation and erosion respectively, which is applied to the point cloud from Figure 6.5. Now with the holes filled in and border noise removed we have a rough shape of the cluster without having to look at every point in the plane.

Implementation Details

For all these steps, dilation and erosion were performed using the 8-neighborhood, as this allows us to fill holes and remove noise with less iterations of dilation and erosion.

Unfortunately, for the dilation step we need to be careful of not running out of the binary image, as the binary image returned from the quad tree is packed very tightly. This means it is necessary to expand the image size temporarily for the dilation process.

The algorithm used for the erosion and dilation process is relatively efficient. The most intuitive approach for multiple Dilation and erosion steps is $O(n^4)$, in which for every pixel we find the closest pixel that is set. So two loops are needed for finding the closest pixel and two are needed to go over all the pixels.

The method we used is based on calculating the Manhattan distance. The Manhattan distance from every pixel to the closest set pixel can be calculated in $O(n^2)$ by using two passes. Starting at the top left pixel and working right and then down we know that the minimum distance is zero if the pixel is set or otherwise the minimum of the four pixels to the left, top left, top or top right. Then the same process is taken again from the bottom right. Now we only have to set all the pixels which have a number smaller or equal than the number of dilation steps to be performed. This is explained in detail in [18].

6.3.3 Border Detection

The last step we performed using the binary image was to find the border. To do so, we eroded the image once using a 4-neighborhood this time, which provided

us with a binary image of which the border has been removed. To obtain the border, we only had to look at all the points pixels in the binary images that were different before and after eroding. This provided us with results like in [Figure 6.9](#).

To be able to draw the points on the border or the polygon filled by the border, we needed to find the actual locations of the points. To do so we found the top left most pixel of the border and walked to the next pixel in a clockwise direction until we reached our starting pixel again. Here we had to make sure that we did not step into a loop somewhere and that we did not accidentally step back to a place that was already visited.

6.4 Polygon Fitting

The results from the previous step were quite good, but the edges were not as straight as they could have been. Since we aimed to reconstruct buildings, straight lines were important for aesthetic reasons.

The method we used here is based on RANSAC [11] again, but this time rather than fitting planes, lines were fitted to the data. When most of the points on the border had been matched to lines, the lines were used and their intersections found. These intersections were then used to define the vertices of the polygon.

This worked very well for good data, but it had problems when one edge of the plane was curved or bent. This led to the system missing one line in a rectangle. Intersecting the three lines that had been found created a triangle rather than the quadrangle that was hoped for.

Off-the-shelf polygon fitting methods are available, like the one found in OpenCV. Unfortunately we did not have enough time left to experiment with these.

Chapter 7

Conclusions

In this project we have developed a partially-automated system that allows users to easily examine thermal images of buildings and identify potential problems. In the process we have consulted both expert and novice users and gathered feedback about what the software is useful for and how it could be improved. We have also managed to incorporate some of these suggestions into our final implementation to make our interface more user-friendly.

Possible extensions to this project include performing more thorough evaluation with a larger number of participants with more diverse backgrounds. Additionally the 3D reconstruction pipeline could be more fully automated. Finally, the system could be made more attractive to the expert users if it included more quantitative information relating to the measurements such as orientation, atmospheric temperature and wind speed.

Based on the feedback we obtained, our system is mostly useful for users who do not have extensive experience and knowledge in building science but need to gain a basic understanding of where buildings lose heat. These include students of architecture and civil engineering.

Bibliography

- [1] MeshLab. <http://meshlab.sourceforge.net/>. 11
- [2] CGAL, Computational Geometry Algorithms Library. <http://www.cgal.org>. 11
- [3] CGAL, Surface Reconstruction from Point Sets. http://www.cgal.org/Manual/latest/doc_html/cgal_manual/Surface_reconstruction_points_3/Chapter_main.html. 11
- [4] Balaras C. A. and Argiriou A. A. Infrared thermography for building diagnostics. *Energy and Buildings*, 34. 5
- [5] A. Bechara and A.R. Damasio. The somatic marker hypothesis: A neural theory of economic decision. *Games and Economic Behavior*, 52(2):336–372, 2005. 19
- [6] J. L. Branch. Investigating the information-seeking processes of adolescents: The value of using think alouds and think afters. *Library & Information Science Research*, 22(4):371–92, 2000. 15
- [7] A.R. Damasio. *Descartes' Error; Emotion, Reason, and the Human Brain*. Avon Books, New York, 1994. 19
- [8] K. Duncker. On problem solving. psychological monographs. *American Psychological Association.*, 58, 1945. 15
- [9] K. A. Ericsson and H. A. Simon. *Protocol analysis: Verbal reports as data*. MIT Press, Cambridge, MA, 1993. 15
- [10] R.A. Finkel and J.L. Bentley. Quad trees: a data structure for retrieval on composite keys. *Acta Inform.*, 4:1–9, 1974. 24
- [11] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Readings in computer vision: issues, problems, principles, and paradigms*, pages 726–740, 1987. 20, 21, 27

- [12] M.E. Fonteyn, B. Kuipers, and S.J Grobe. A description of think-aloud method and protocol analysis. *Qualitative Health Research*, 3(4):430–441, 1993. 15
- [13] Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Clustering views for multi-view stereo. <http://grail.cs.washington.edu/software/cmvs>. 9
- [14] Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Towards internet-scale multi-view stereo. In *CVPR*, 2010. 9
- [15] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2009. 9
- [16] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004. 7, 9
- [17] D. R. Myatt, P. H. S. Torr, S. J. Nasuto, J. M. Bishop, and R. Craddock. Napsac: high noise, high dimensional robust estimation. In *In BMVC02*, pages 458–467, 2002. 21
- [18] Stephen Ostermiller. Efficiently implementing dilate and erode image functions. http://ostermiller.org/dilate_and_erosion.html. 26
- [19] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. In *ACM TRANSACTIONS ON GRAPHICS*, pages 835–846. Press, 2006. 9
- [20] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections, 2007. 7, 9, 11
- [21] James Tompkin. Script combining bundler, cmvs and pmvs, 2010. 9
- [22] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCV '99: Proceedings of the International Workshop on Vision Algorithms*, pages 298–372, London, UK, 2000. Springer-Verlag. 9
- [23] M. W. Van Someren, Y. F. Barnard, and J. A. C. Sandberg. *The think aloud method: A practical guide to modeling cognitive processes*. Academic Press, London, 1994. 15