NTNU
Det skapende universitet

Analyse av algoritmer for fotogrammetrisk generering av punktskyer

John Herman Wika Haakseth

Master i ingeniørvitenskap og IKT
Innlevert: juni 2013
Hovedveileder: Knut Ragnar Holm, BAT
Medveileder: Leif Erik Blankenberg, Blom Geomatics

Norges teknisk-naturvitenskapelige universitet
Institutt for bygg, anlegg og transport

Problem description

Analysis of Algorithms for Photogrammetric Generation of Point Clouds

The assignment involves comparing the quality of point clouds generated by different software packages/algorithms, as well as investigating how adjusting the parameters of the algorithms affects the result. It is also of interest to shed light on the effect of the aerotriangulation on the final result.

Supervisors

- Professor Knut Ragnar Holm, *Norwegian University of Science and Technology*
- Dr. Ing. Leif Erik Blankenberg, *Blom Geomatics*

Abstract

This thesis presents a study in which three different systems that perform dense image matching on multiple aerial photography to generate point clouds are assessed. Two approaches of analyzing them were explored. The first one gives descriptive statistics and visual characteristics about each output point clouds with respect to itself. The second approach is a comparative analysis in which the point clouds are assessed with respect to the point cloud of a LIDAR scan covering the same area.

Chapter 2 gives an introduction to image processing and basic filtering techniques for image enhancement and feature detection. Traditional area-, and feature-based algorithms for image matching are then explained, before the modern trends like cost-based and semi-global matching are presented.

The results of the comparative analysis showed that, in the urban areas, Match-T DSM was the most accurate, while UltraMap was the most stable with respect to both accuracy and precision for all areas. Socet Set NGATE generates the most visually pleasing result, but it seems to contain more blunders in the output height data, despite rejecting uncertain points in the matching process.

Sammendrag

Denne avhandlingen presenterer en studie der tre forskjellige systemer som utfører dense bildematching på flyfoto for å generere punktskyer vurderes. To tilnærminger for å analysere dem ble utforsket. Den første gir beskrivende statistikk og visuelle karakteristikker om hver resulterende punktsky. Den andre metoden er en komparativ analyse hvor punktskyene blir sammenlignet med en punktsky fra en LIDARskanning som dekker det samme område.

Kapittel 2 gir en innføring i bildebehandling og grunnleggende filtreringsteknikker for bildeforbedring og objektgjenkjenning. Deretter blir tradisjonelle areal- og objektbaserte algoritmer for bildematching blir gjennomgått, før moderne trender som kostnadsbasert og semiglobal bildematching presenteres.

Resultatene av den komparative analysen viste at i urbane områder, var Match-T DSM den mest nøyaktige, mens UltraMap var den mest stabile med hensyn til både nøyaktighet og presisjon for alle områdene i studien. Socet Set NGATE genererer de mest visuelt tiltalende punktskyene, men de ser ut til å inneholde flere grove feil i høydeverdiene sammenlignet med de andre systemene. Dette til tross for at NGATE utelukker usikre punkter i prosesseringen.

Preface

There are many people to thank for their input during this study. The first are naturally my supervisors, Knut Ragnar Holm and Leif Erik Blankenberg, who have provided valuable feedback, advice and literature. Also, the practical and professional help and training I received from Blom Geomatics, with Floris Jan Groesz, Bernt Larsen, Inge Myklebust and Ivar Oveland, has been invaluable.

Finally, I would like to thank all my fellow students and the faculty staff at Geomatics at NTNU for three great final years at NTNU. A special thanks to my dear colleague Stine Røsjorde Lund. To me, she has been a pillar of support throughout these years, academically, and spiritually.

Copyrighted material

Table of Contents

Problem Description i

Abstract iii

Sammendrag v

Preface vii

Table of Contents xi

List of Tables xiii

List of Figures xv

I Preface 1

1 Introduction 3
1.1 Background and motivation 3
1.1.1 Image matching 3
1.2 Scope of thesis 4

II Theory and methodology 5

2 Image processing and matching 7
2.1 Introduction 7
2.2 Digital representation of an image 7
2.3 Filtering 9
2.3.1 Applying a filter to an image 9
2.3.2 Handling edge of image 10
2.3.3 Types of image filters 10

2.4 Image matching . . . 15
2.4.1 General strategy . . . 15
2.4.2 Area-based image matching . . . 15
2.4.3 Feature based image matching . . . 18

3 Materials and methods **19**
3.1 Study area . . . 19
3.1.1 Oslo S . . . 20
3.1.2 Urtegata . . . 20
3.1.3 Botanical garden . . . 20
3.2 Datasets . . . 21
3.2.1 Aerial imagery . . . 21
3.2.2 LIDAR dataset . . . 21
3.2.3 Height reference systems . . . 23
3.3 Software packages . . . 24
3.3.1 Match-T DSM . . . 24
3.3.2 Socet Set NGATE . . . 26
3.3.3 UltraMap . . . 26
3.4 Descriptive statistics and visual characteristics of point clouds . . . 27
3.5 Comparative analysis . . . 28
3.5.1 Workflow . . . 28

III Result **31**

4 Descriptive statistics and visual characteristics of point clouds **33**
4.1 Introductory remarks . . . 33
4.2 Match-T DSM . . . 34
4.3 Socet Set NGATE . . . 36
4.4 UltraMap . . . 38

5 Comparative analysis **41**
5.1 Oslo S . . . 41
5.2 Urtegata . . . 41
5.3 Botanisk hage . . . 42

IV Discussion and conclusions **45**

6 Discussion **47**
6.1 Visual characteristics of the resulting photogrammetric point clouds . . . 47
6.2 Comparative analysis . . . 47
6.3 Image matching vs. laser data . . . 49
6.4 Effect of aerotriangulation . . . 50

7 Conclusions and further work **53**
7.1 Conclusions 53
7.2 Further work 54

References **55**

Appendix **59**

List of Tables

2.1 Representation of pixel values in an image 8

3.1 Key information, aerial photography data set 22
3.2 Key information, LIDAR data set 23
3.3 Match-T DSM settings used in the comparative analysis 26
3.4 Socet Set NGATE settings used in the comparative analysis 27
3.5 UltraMap settings used in the comparative analysis 27

4.1 Profile analysis result 33
4.2 Match-T DSM key features 35
4.3 Socet Set NGATE point cloud key features 37
4.4 UltraMap point cloud key features 38

5.1 Summary, DSM subtraction, Oslo S 42
5.2 Summary, DSM subtraction, Urtegata 42
5.3 Summary, DSM subtraction, Botanisk hage 43

List of Figures

2.1 Combining red, green, and blue band to achieve color image 8
2.2 Moving average filter 9
2.3 Applying a sharpening filter to image. 12
2.4 Effects of gradient edge detection filters 13
2.5 Smoothing image before edge detection 14
2.6 Template and search window 16

3.1 Overview of study area 19
3.2 Areas for comparative analysis 20
3.3 Oslo S 21
3.4 Urtegata 22
3.5 Botanical Garden 23
3.6 Coverage of aerial imagery 24
3.7 Laser data coverage 25
3.8 Profile analysis tool 28

4.1 Profile analysis comparison of point clouds 34
4.2 Point cloud from Match-T 35
4.3 Alternative parameter settings, Match-T 36
4.4 Ungridded vs. gridded point cloud from Socet Set 37
4.5 Visual impression of point cloud from UltraMap 38
4.6 UltraMap pointcloud height variability 39

6.1 Accuracy and precision 48
6.2 Ground Control Points in the study area 51

Part I

Preface

Chapter 1

Introduction

1.1 Background and motivation

For the past 10 years, LIDAR scanning has been the market leading technology for sensing accurate 3D terrain data, with applications in, among others, aerial photogrammetry for forestry and urban areas and terrestrial sensing for transportation and architecture. Consequentially, we have seen much innovation and research interest on LIDAR technology and processing in this period [1].

Many applications in mapping and remote sensing have still had a need for aerial photography. In the past decade, aerial imagery has seen a complete transformation from a traditional analog workflow, to a completely automated digital one. In turn, this has meant that accumulating more data is cheaper, as more can be stored on board (the airplane) and the processing of the images is faster.

The fact that more data can be sensed and processed means that a higher (forward) overlap in images can be collected at no extra cost. This allows for less occluded areas in the images and more redundancy in the data sets. Researchers and commercial companies have taken interest in this opportunity and recently photogrammetric software packages have started to include modules for image matching and creating output comparable to LIDAR scans, at pixel or even sub-pixel level.

1.1.1 Image matching

The general problem of image matching can be defined as the task of matching two or more pictures which have been taken of the same object or scene, from e.g., different angles or with different sensors. For photogrammetric purposes these are overlapping images acquired from airborne sensors. Matching points and features between the images are then used to measure three-dimensional coordinates that can be transformed into a geographical coordinate system.

Techniques for image matching have been a research topic for almost 50 years, but the general problem remains unsolved [2]. These techniques have gotten more attention

in recent years as the acquisition of imagery has gotten significantly easier and cheaper as digital photography became the norm.

The topic of image matching is presented in Chapter 2.

1.2 Scope of thesis

This study will present an introduction and background to the theory and algorithms used for image matching. Subsequently, a comparison of results from three different photogrammetric software packages is presented. The softwares that were compared are all proprietary and a specific presentation of the algorithms used within each of them, outside of what is available in literature, is not a part of this study.

This study is not a comparison of results from image matching techniques and LiDAR captured data. A comparative analysis is presented in this report and the LiDAR data set is used in order to have a common reference to compare each software against.

Part II

Theory and methodology

Chapter 2

Image processing and matching

2.1 Introduction

For photogrammetric purposes, image matching has been a research topic since the 1980s. Complex algorithms and lack of computing power has prevented it from gaining traction as a viable method for photogrammetric purposes. As more powerful computers have gotten accessible, research interest in this topic has reignited [1], and as a result, commercial software packages have started including modules for creating dense point clouds from airborne image data.

As the available software packages are commercial and closed source, knowing exactly what they do, is difficult. This chapter will therefore present common methods and algorithms for image matching. Matching techniques are generally classified into two categories, *area based* and *feature based* image matching [3]. These form the basis for the image matching section of this chapter. Before presenting the algorithms, some of the basics of digital image representation and processing of them are explained.

2.2 Digital representation of an image

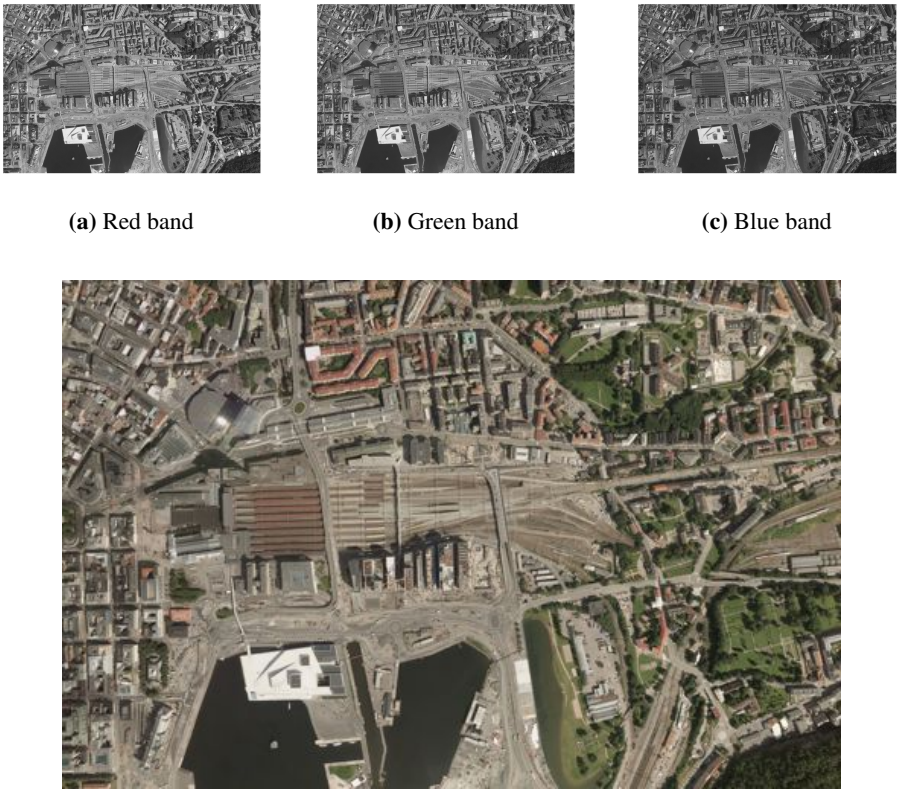
To analyze and process digital images, one has to know how they are represented and interpreted by a computer. This presentation will be limited to two-dimensional images. In such cases an image can be represented by

$$f(x, y) \tag{2.1}$$

where x is the column, y is the row and then f is the pixel value at x, y. This value represents the intensity of the sensor band, and is typically represented as an 8-bit integer. For a single band image (e.g. grayscale), this can be represented in a two-dimensional grid as in Table 2.1, where each value is the intensity of that pixel. A single band image is usually represented as a grayscaled image, as seen in Figure 2.1 a)-c). For multi band images (e.g. RGB) we have one of these for each band, as illustrated by Figure 2.1.

233	220	220	175	153	145	155	155	28	25	29	155
240	219	176	163	127	83	77	30	23	27	59	104
245	239	209	176	134	98	24	20	23	89	134	185
231	234	202	156	120	43	25	30	33	120	187	201
243	202	177	123	56	24	24	26	52	129	181	236
200	212	152	132	37	30	29	36	72	137	198	219
176	139	198	67	34	26	32	40	92	163	198	247
184	191	120	92	33	30	35	99	102	137	195	237
165	86	26	22	21	29	56	102	142	192	204	230

Table 2.1: Table representation of values for individual pixels of an 8-bit grayscale image, or more generalized as a single-band image.



(a) Red band **(b)** Green band **(c)** Blue band

(d) Colored RGB-image.

Figure 2.1: Combining different bands, typically red, green, and blue, lets the computer display them on a screen and let the viewer interpret them as natural colors.

2.3 Filtering

Filtering [4], also called convolution, of digital images is a processing method where a *kernel*, or convolution matrix, is applied to each pixel in the image. The idea is that the neighboring pixels influence the center one and outputs a different image. From the output image, information may be easier to extract than from the original one.

This can be a useful first step of extracting features before comparing images for image matching that will be presented in Section 2.4.3.

2.3.1 Applying a filter to an image

The calculations are done by applying weights to the center pixel and each of its closest k (depending on kernel size) neighbors and normalizing them. In Figure 2.2 an averaging filter (right) can be applied to the center pixel (left) by the following calculation:

$$\frac{1}{9}(233 + 220 + 220 + 240 + 219 + 176 + 245 + 239 + 209) \approx 222 \tag{2.2}$$

233	220	220
240	219	176
245	239	209

1	1	1
1	1	1
1	1	1

Figure 2.2: Applying the simple moving average filter to the right to the center pixel in the left, yields 222 after normalizing, as seen in Equation 2.2.

The sum of the weights in the kernel is 9. To ensure that the output pixel value is of the same magnitude as the input, we divide by the sum of weights (normalizing).

A general, normalized $3 * 3$ kernel can be represented on the following form:

$$\frac{1}{sum(w)} \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}, sum(w) = \sum_{i=1}^{9} w_i \tag{2.3}$$

Where w_i represents the weights applied to each pixel. Depending on how many of the nearest k neighbors we want to influence the new value of the center pixel, the kernel size can vary. For kernels with equal number of rows and columns, the kernel size will be:

$$n = 2k + 1 \tag{2.4}$$

Where the kernel will be of size n rows and n columns. The size of the kernel will always be odd-numbered.

2.3.2 Handling edge of image

When applying kernels to images, how to handle the edge of the image must be taken into consideration. Three different strategies exist:

Extending the boundary values: The pixel value beyond the image boundary is given the same value as the nearest edge pixel.

Wrapping or tiling the values: The pixel value beyond the image boundary is given the same value as the pixel on opposite end of the image, as if the image was tiled.

Cropping image: Cropping image to exclude pixels with too few neighbors. Output image will be $n - 1$ rows and columns smaller than input.

2.3.3 Types of image filters

Three common types of filters are used in image processing:

1. Smoothing
2. Sharpening
3. Edge detection

The type of filter is determined by the weights given in the kernel. To introduce the kernel types, a kernel that leaves the image unchanged is presented first:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.5}$$

Smoothing filter

A smoothing filter, also called averaging, blurring, or low-pass filter, is a filter that smooths out rapid changes in the pixel values. A simple smoothing filter is the moving average filter, shown in Figure 2.2. The normalized version of this can be shown as:

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{2.6}$$

These kinds of filters are commonly used for noise reduction or preprocessing images for land use classification purposes.

Sharpening filter

A sharpening, filter works in the opposite way compared to the smoothing filter. If there is a change from one pixel to the next, this filter will emphasize the difference. Subtracting a low-pass filter from the filter that does nothing, yields a high-pass filter:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{9}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{2.7}$$

Adding the resulting image to the original will result in a sharpened image. This technique enhances edges, but will also cause more noise in the output image. An example of this can be seen in Figure 2.3.

Edge detection filter

Edge detection filters can be classified into gradient, line detection and laplacian filters [5]. Gradient filters look for edges in a specified direction. For kernels of size 3, these can be defined in $45\,^{\circ}$ increments for example:

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.8}$$

These filters will look for changes in the east, north-east and north direction, respectively. The results of which are shown in Figure 2.4.

Similar to gradient filters, are line detection filters:

$$\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}, \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}, \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \tag{2.9}$$

These look for horizontal, diagonal, and vertical lines, respectively.

Laplacian filters look for edges in all directions at once. Example filter:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{2.10}$$

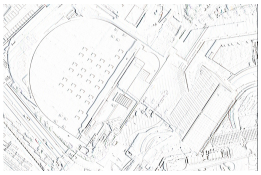
Applying smoothing before this filter is applied to an image will reduce impact of noise, e.g. Canny-filter [6, 7]. The effect of this can be seen in Figure 2.5. Many image matching algorithms consist of a feature (e.g edge) matching component, as presented in Section 2.4.3.

(a) Original image.

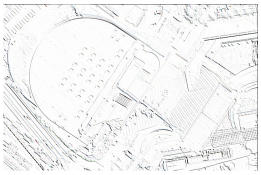
(b) Sharpened image.

Figure 2.3: Applying a sharpening filter to image.

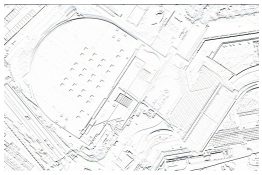
(a) Original image, with smoothing filter applied to reduce noice.



(b)



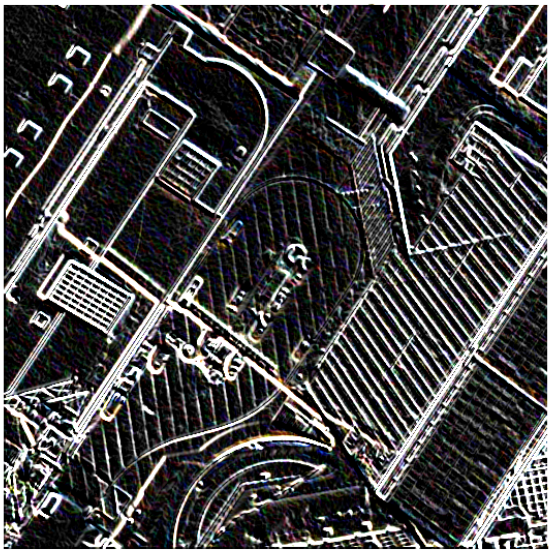
(c)



(d)

Figure 2.4: Input image (a), east gradient filter (b), north-east gradient filter (c), and north gradient filter (d).

(a) Only edge detection filter



(b) First smoothing filter, then edge detection

Figure 2.5: Applying smoothing to image before searching for edges may reduce noise.

2.4 Image matching

This study focuses on overlapping images captured from airborne sensors. The problem of image matching can thus be stated as the following, paraphrased from [3]:

> Given a *reference* image, $I_1(x, y)$, and a *input* image, $I_2(x, y)$, find the geometric transformation, $f(g(x, y))$, that "best" transforms I_2 onto I_1.

This definition only takes *geometric* transformation into account, in most photogrammetric applications there will also be a *radiometric* transformation. Area-based and feature-based techniques have been the leading approaches for photogrammetric image matching for over 20 years [2]. These techniques have some limitations, however. Feature-based methods of image matching can only match distinct features that it finds, making it a *sparse* matching technique. Area-based methods will suffer from a smoothing effect based on the window sizes used. This effect makes it difficult to match buildings with sharp edges accurately. There are, however, some new image matching techniques that try to overcome these issues.

Semi-Global Matching (SGM) [8] is an algorithm that attempts to perform one match per image pixel. The technique involves relating each pixel in one image to each in the other with respect to a matching cost. The sum of all costs defines a global matching cost which is assumed to be minimal for the best solution. One method of calculating cost is by *mutual information*. The mutual information of the two images can be defined as the entropy of each image, as well as their joint entropy:

$$MI_{I_1,I_2} = H_{I_1} + H_{I_2} - H_{I_1,I_2} \tag{2.11}$$

The following sections will present a general image matching strategy, and the two main approaches of image matching, area-based matching (ABM) and feature-based matching (FBM).

2.4.1 General strategy

The majority of image matching strategies consists of the following steps [9]:

1. Feature detection
2. Feature matching
3. Transform model estimation
4. Image resampling and transformation.

2.4.2 Area-based image matching

Area-based (also called template-based) image matching algorithms are applied to single-band images and is used for two images at the time [10]. The basic idea is that a small template window is defined in the first image (the template window) and find the corresponding area within a larger window in the second image (the search window), as shown

in Figure 2.6. For each pixel row and column of the search window, the template window is shifted over and a similarity metric is calculated. The basic algorithm is described in Listing 2.1.

Listing 2.1: Pseudocode of area based image matching

```
for each x,y in searchWindow:
  for each u,v of templateWindow:
    do compare template and search window
    if(this match is best so far):
      store match as bestMatch
      store current x,y as bestPos
    end if
  end for
end for
return bestMatch, bestPos
```

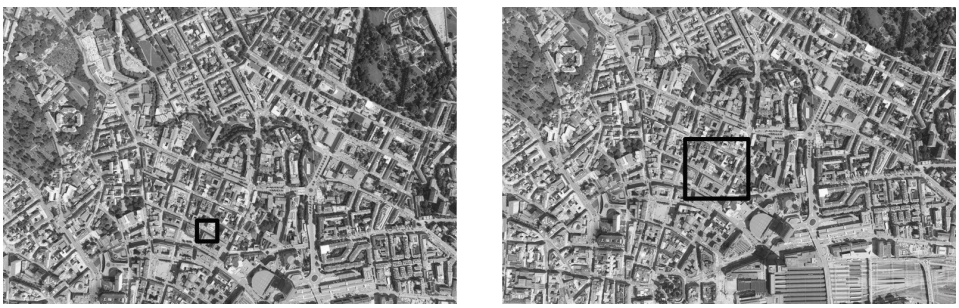


Figure 2.6: Two overlapping images with template window (left) and search window (right).

What separates different algorithms from each other, is how the similarity metric is decided. The following section presents two common variations.

Cross-correlation

The similarity criteria can be the correlation between the template window, $f(x, y)$, and the search window with R rows and C columns, $g(x, y)$. The cross-correlation between them is thus [10]:

$$\rho(t_r, t_c) = \sum_{r=0}^{r=R-1} \sum_{c=0}^{c=C-1} f(r, c) g(r + t_r, c + t_c) \tag{2.12}$$

This algorithm is fairly simple to understand, but has limits. It assumes that the illumination is the same for both images. This is usually not the case. To eliminate the problem of differences in illumination, the normalized cross-correlation coefficient can be

used. This is done by subtracting the mean intensity value in each window from each intensity value and dividing the root squared sums of the reduced intensity values in the two windows [3, 10]:

$$NCC_{f,g}(x,y) = \frac{\sum_{r=0}^{r=R-1}\sum_{c=0}^{c=C-1}[f(r,c)-\bar{f}][g(r+t_r,c+t_c)-\bar{g}]}{\sqrt{\sum_{r=0}^{r=R-1}\sum_{c=0}^{c=C-1}[f(r,c)-\bar{f}]^2\sum_{r=0}^{r=R-1}\sum_{c=0}^{c=C-1}[g(r+t_r,c+t_c)-\bar{g}]^2}} \tag{2.13}$$

The normalized correlation method will yield values $\rho \in (0,1]$ Often a threshold value is set prior to the search, and a match is considered successful if the cross-correlation exceeds this value, e.g. $\rho > 0.9$.

Least Squares Area-Based Matching

Like the cross-correlation method, Least Squares Matching (LSM) uses a template and search window to compare grey values [10]. Compared to the cross-correlation method, LSM has some advantages:

Geometric transformation: In addition to just translation, LSM can model scale differences and rotations.

Precision: Under good conditions this method can yield accuracy of up to $0.01-0.02*$pixel-size.

Error estimation: The method can produce an estimate for the errors in the transformation parameters.

The basic idea of LSM is to minimize the differences of intensity between the template and search window. To do this, the algorithm tries to find the best possible position (translation), as well as the shape (rotation and scale) of the matching window. [11] gives a description of the procedure:

- An initial approximate transformation of the template window onto the search window is found (e.g. by cross-correlation or feature-based matching).
- Intensity values are compared, allowing for constant and linear changes in relation those in the second image.
- The sum of squared differences of the corresponding normalized intensity values are minimized, based on the principle of least squares, using the transformation parameters as unknowns:

$$v(x,y) = g(x,y) - [r_0 + r_1 * g'(x',y')] \tag{2.14}$$

$$x' = a_0 + a_1 x + a_2 y \tag{2.15}$$

$$y' = b_0 + b_1 x + b_2 y \tag{2.16}$$

Where:

- x, y and x', y' represent the image coordinates of the first and second image.
- $g(x, y)$ and $g(x', y')$ represent the intensity values in the first and second image.
- r_0 and r_1 are the intensity normalization parameters.
- $a_0, a_1, a_2, b_0, b_1 and b_2$ are the affine transformation parameters.
- $v(x, y)$ are the residuals of normalized intensity value differences.

One strategy in LSM is to give the algorithm a seed starting point (a corresponding point for both images). The algorithm can then try to match in the four neighboring points, and continue in the point with the highest correlation coefficient. A well-adjusted threshold for this window is essential for the success of the algorithm.

This strategy is called region growing LSM. After a while, the algorithm will stop, when it can't find a high enough correlation. It then needs more seed points to continue. The accuracy of the method will depend on the window size. If the windows are large, the accuracy will be high, but there will be less detail.

2.4.3 Feature based image matching

Area-based image matching is sensitive to changes in illumination and perspective between the different images. This can lead to matching errors, and many systems employ a feature matching step before employing an area based method to account for large differences

Basic procedure of feature based image matching [12]:

1. Extract basic features (patches, corners, junctions, edges, etc).
2. Build up a list of candidate paris of features based on a similarity measure.
3. Derive final list of feature pairs consistent with an object model.

One way to extract features is by edge detection, as explained in Section 2.3.3. Other examples of features are corners, blobs (interest operators) [13]. The most common way to match the candidate features is by area-based matching:

1. A template is placed around the feature to be matched.
2. There may be several possible features within the search window. For all of these the correlation coefficient is calculated, and the highest is considered to be the match.
3. Repeat 1 and 2 for all features.

Intuitively, this is faster than the methods described above, in Section 2.4.2, but will only yield matches for the features found. This is why feature-based matching often is used as a first step in other image matching techniques.

Chapter 3

Materials and methods

3.1 Study area

The study area is located in the Bjørvika area of central Oslo, Norway. The land cover is mainly urban, with some green areas to the north. An overview of the study area can be seen in Figure 3.1.

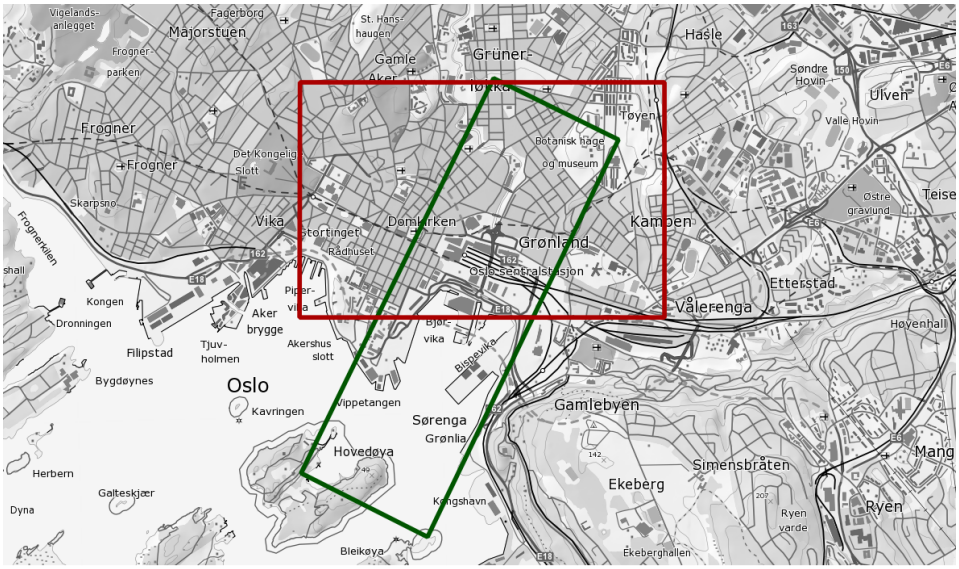


Figure 3.1: Overview of study area, with polygons showing the coverage of LIDAR data (red) and aerial imagery covered by both flight strips (green).

Three areas within the study area were selected for a comparative analysis, as presented in Chapter 5. These were selected on the basis of their varying topography and land cover,

as well as their location within the area that has both LIDAR data and aerial imagery. The location of these areas within the study area can be seen in Figure 3.2.

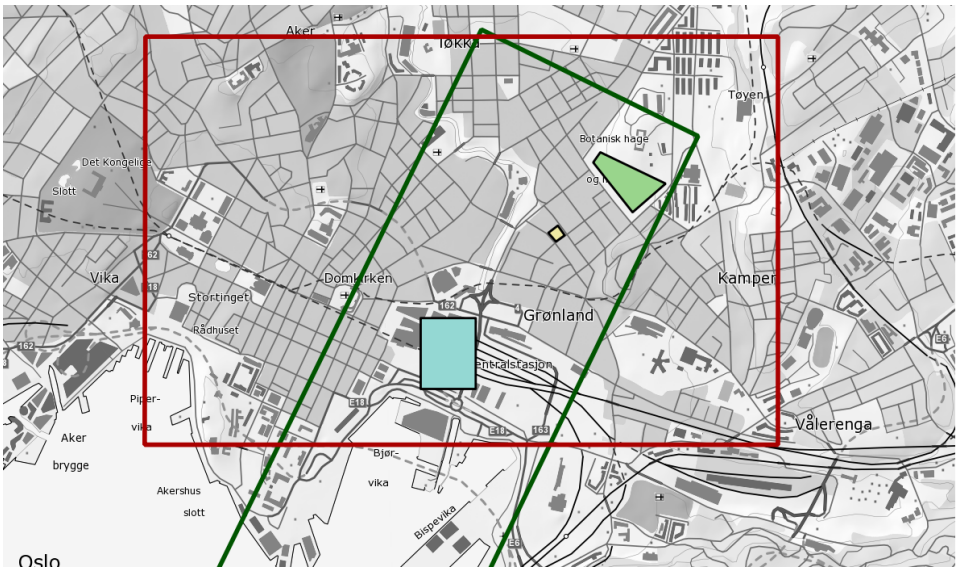


Figure 3.2: Polygons showing the areas for comparative analysis, seen inside the study area. Blue: Oslo S, Yellow: Urtegata, Green: Botanisk hage.

3.1.1 Oslo S

Oslo S is the train station in central Oslo, located in the center of the main study area, and in the bottom, southern part of the area with coverage of both LIDAR data and aerial imagery. The clipped area consists of buildings, rail tracks, roads, some vegetation and other complex structures, as seen in Figure 3.3.

3.1.2 Urtegata

Urtegata is a street in the center of the study area. The clipped area is limited to a group of rooftops, as seen in Figure 3.4.

3.1.3 Botanical garden

The Botanical Garden is the green area to the north of the study area. The clipped area is characterized by a gently sloped area mostly covered with grass and trees, as seen in Figure 3.5.

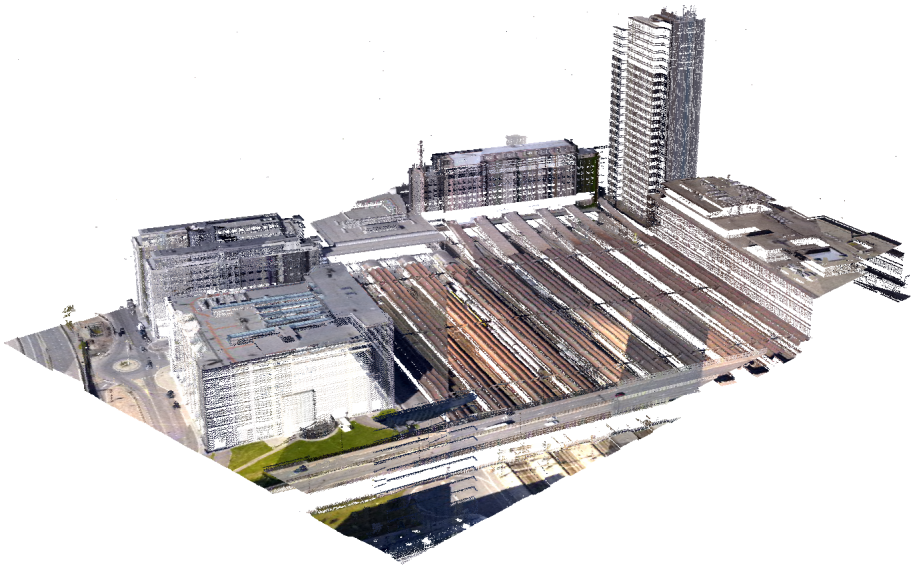


Figure 3.3: The point cloud of Oslo S, from the LIDAR dataset.

3.2 Datasets

3.2.1 Aerial imagery

The input aerial imagery were captured by Blom Geomatics using Vexcel UltraCam Xp sensors. This was part of a test project utilizing high overlap. Table 3.1 presents key information about the image dataset. Two flight lines were flown, each $4.4\mathrm{km}$, with a forward overlap of 80% and a side overlap of 60%, resulting in a area of approximately 3300m by 1100m that is covered by both flight lines, as seen in Figure 3.6.

Post-processing of the imagery, including import, radiometry adjustments, aerotriangulation (AT) were all performed in the UltraMap framework. The result of the AT was exported to a BINGO project [14]. The BINGO format is supported by all the softwares in this study, and ensured they all had the same basis for the image match.

3.2.2 LIDAR dataset

The control datasets are laser scanned datasets, captured by low-flying helicopters. This dataset is owned by Oslo Municipality. The project originally covered more of the Oslo area, but for the purpose of this study, only data that also had image coverage were used. Coverage of the LIDAR data used in this study, can be seen in Figure 3.7. Table 3.2 presents key information about the LIDAR dataset.

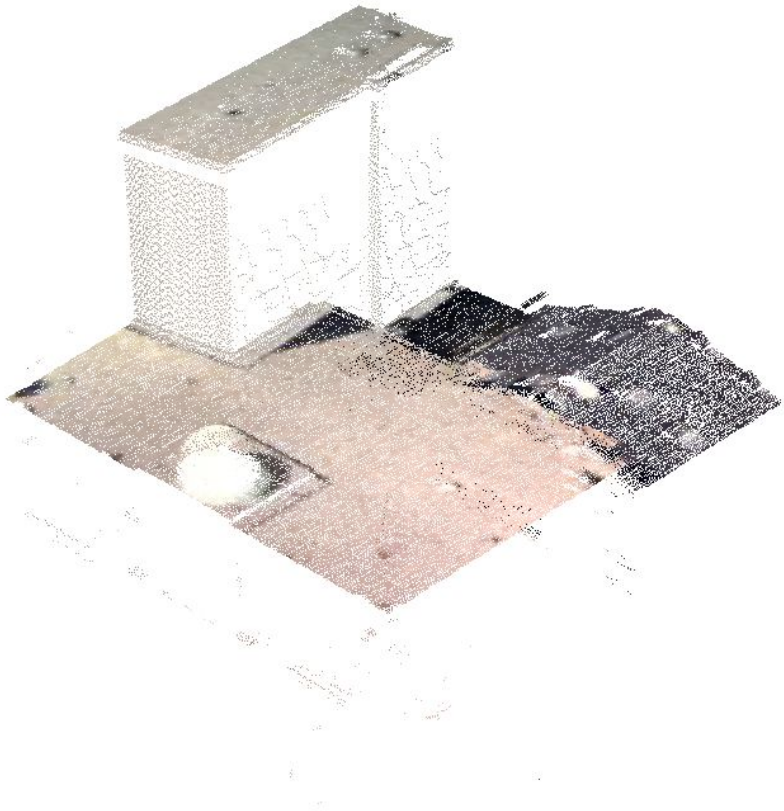


Figure 3.4: The point cloud of the rooftops at Urtegata, from the LIDAR dataset.

Key information, aerial photography data set	
Format	JPEG, RGB.
Projection	WGS 84, UTM32
Height reference	NN2000
Flight altitude	$1670\mathrm{m}$
GSD	$10\mathrm{cm}$
No of exposures	22, in 2 flight lines
Average image size	$105\mathrm{MB}$, compressed with factor Q4, lossless
Capture date	2011

Table 3.1: Key information, aerial photography data set

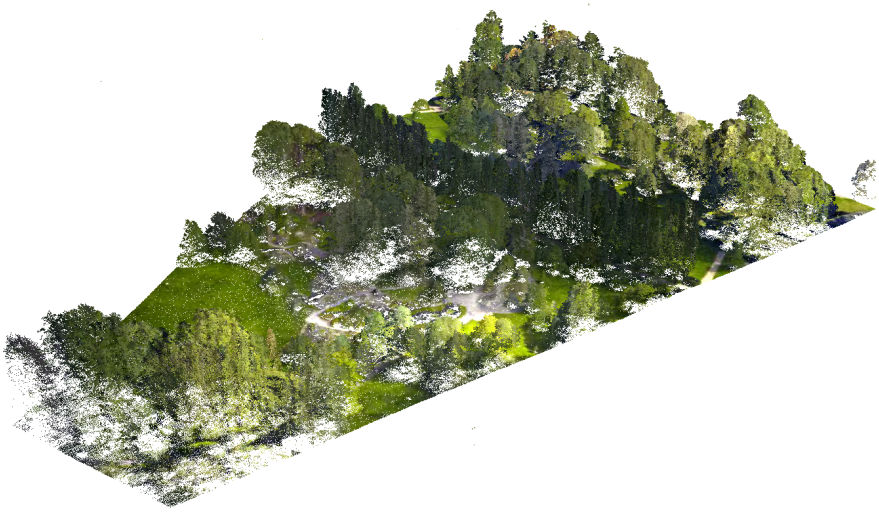


Figure 3.5: The point cloud of the Botanical Garden, from the LIDAR dataset.

Key information, LIDAR data set	
Format	LAS 1.2.
Projection	WGS 84, UTM32
Height reference	NN2000
Flight altitude	$850\mathrm{m}$
No of LAS tiles	42 (used in this study)
Average LAS tile size	$193\mathrm{MB}$
Point cloud density	$\sim 73\frac{\mathrm{Pt}}{\mathrm{m}^2}$
Capture date	2011

Table 3.2: Key information, LIDAR data set

3.2.3 Height reference systems

As seen in Table 3.1 and 3.2, the datasets are in different height reference systems. The aerial imagery has been aerotriangulated using reference points in the height reference system used by Oslo Municipality, *Oslo Lokal*. The LIDAR dataset was processed using the new official height system of Norway, *NN2000*.

To find the correction factor between the height reference systems within the study area, the WSKTrans software [15] was used. WSKTrans is able to perform transformations between (among others) the old Norwegian height reference system, *NN54* and NN2000. First, the height values in Oslo Lokal needed to be transformed to NN54. The following correction was applied [16]:

$$\mathrm{NN54} = \text{Oslo Lokal} + 21.2\mathrm{cm} \tag{3.1}$$

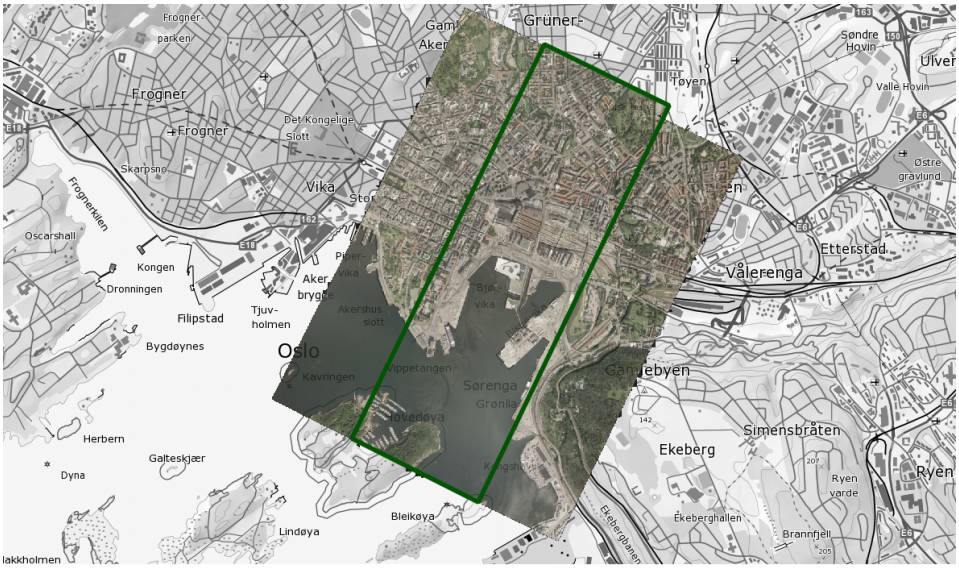


Figure 3.6: Mosaic showing the area that is covered by the aerial imagery overlaid a topographic map of Oslo. The green polygon indicates the area with sideways overlap.

Using NN54 values as input in WSKTrans, the corresponding values of the reference points were found to be 37cm higher in NN2000 than Oslo Lokal (for the study area).

$$\text{NN2000} = \text{Oslo Lokal} + 37\text{cm} \tag{3.2}$$

Note that this is not a general transformation between the height reference systems. It is only valid within the study area. The transformation was applied to the datasets in the comparative analysis, as noted in Section 3.5.

3.3 Software packages

This section will present the photogrammetric softwares that were used in this study. Three different softwares were compared. Each section will present what image matching strategy can be found in literature and the settings that were used in the comparative analysis of this study.

3.3.1 Match-T DSM

Match-T DSM 5.5 is a part of Inpho's ApplicationsMaster suite [17], which is a set of photogrammetric tools for import, export, aerotriangulation and editing of image and DTM data. The documentation of Match-T [18] lists three main image matching algorithms:

Feature Based Matching: FBM was presented in Section 2.4.3. Match-T uses this as a first step to establish a basis for the matching process.

Figure 3.7: LIDAR data coverage.

Least Squares Matching: LSM was presented in Section 2.4.2. After the FBM-step, Match-T uses a template of $21 * 21$ pixels to look for matches in the search window. This process is done iteratively until the square-root of square-sum of gradient residuals converge to a minimum.

Cost Based Matching: Finally, a CBM step computes the high-density surface models. This method is related to the Semi-Global Matching algorithm that was briefly introduced in to Chapter 2.

The following parameter settings may impact the resulting point cloud in Match-T DSM and Table 3.3 shows the settings that were used for the point cloud in the comparative analysis.:

Generation type: Sets default values for further parameters. DTM and DSM possible.

Terrain type: Sets default values for further parameters. Urban, mountainous,

Smoothing: Higher smoothing factor result in less dense point cloud and vice versa.

Feature density: Regulates the amount of ground points that are automatically matched in the DTM generation process.

Parallax threshold: Defines a search area for corresponding points in overlapping images. For regions with large height differences (e.g. buildings) a larger parallax threshold is needed.

Match-T DSM Settings	
Generation type	DSM
Terrain type	Flat
Smoothing	Low
Feature density	Medium
Parallax threshold	10 pixels

Table 3.3: Match-T DSM settings used in the comparative analysis

3.3.2 Socet Set NGATE

Intergraph's Socet Set is a software suite designed for photogrammetric tasks. NGATE stands for Next-Generation Automatic Terrain Extraction, and is a module for automatic extraction of heights from aerial imagery [19].

The literature reveals few specific details about NGATE's image matching strategy. According to [20, 19], Socet Set NGATE's algorithm uses a combination of area-based matching and edge-based image matching. The result from the edge-based are used to assist with the feature-based matching, and vice versa. The results are then combined with algorithms for blunder detection and inconsistency checking.

The workflow of generating point clouds in NGATE is presented in [19]. Table 3.4 shows the settings that were used for the point cloud in the comparative analysis. The settings that impact the resulting point cloud are:

X- and Y-spacing: Decides the density of the output point cloud, was set to 10cm in this study.

Format: TIN triangles or grid. If grid is selected, a height value will be set according to the X- and Y-spacing. For TIN, the algorithm will only set a height value if a good quality match is found.

Strategy: Two strategies are available. "ngate" is intended for small to medium scale imagery, and "ngate_urban" is for large scale imagery.

Maximum number of image pairs per point: If more than one image pair is available, [19] recommends two or three.

Eliminate trees/buildings/other: If this is on, NGATE will generate a DTM, while it will generate a DSM if it is off.

Precision/Speed: Sets which minification level the process should stop on.

TIN Masspoints: Can be set to no, medium, or heavy thinning. This should be set to medium or heavy if the user desires a smaller data volume.

3.3.3 UltraMap

UltraMap 3.0 [21] is a software suite for a photogrammetric workflow, designed to work with imagery from Vexcel's UltraCam sensors. Its framework consists of modules for

Socet Set NGATE Settings	
Spacing (X & Y)	10cm
Format	TIN
Strategy	ngate_urban
Image pairs per point	3
Eliminate trees/buildings/other	Off
Precision/Speed	High/Slow

Table 3.4: Socet Set NGATE settings used in the comparative analysis

importing, performing radiometry adjustments and aerotriangulation of the imagery. Recently modules for dense image matching and ortho mosaic creation were added [22].

Few specifics are given as to the image matching strategy in the literature. Reitinger [23] cites Hirschmüller's [24] semi-global matching strategy, introduced in the image matching part of Chapter 2, and Klaus, Sormann and Karner [25] as comparable approaches. He goes on to say that "image-based correlation methods, (e.g. normalized cross correlation)" are used to establish correspondences between image pairs. The next step, he says, is to perform a range image fusion to calculate a 3D representation (point cloud).

The UltraMap workflow provides very few user-selectable inputs [26]. The settings chosen in the comparative analysis is shown in Table 3.5.

UltraMap Settings	
DSM Processing mode	Robust
Cell size (Vertical/Horizontal)	10cm/10cm
RGB Color	Yes

Table 3.5: UltraMap settings used in the comparative analysis

As the aerial imagery is processed in the UltraMap framework, creating the point clouds is the next step in the integrated workflow there. The process consists of two steps that allows user inputs. First, a DSM is created from the imagery in the Dense Matcher module. This module has one user input that may affect the matching in a qualitative manner, and this is **DSM Processing Mode**, where the user may select *Standard* and *Robust*.

In addition the user can set parameters for computing power, which will affect how long the process will take. When the DSM generation is complete, the process continues in the Ortho Pipeline module. Here the user can view the DSM created in the last step and select tiles for point cloud export.

3.4 Descriptive statistics and visual characteristics of point clouds

Chapter 4 presents each matching result individually. The chapter is organized by software package, and a description of the visual characteristics of each point cloud is given. In

addition, key statistics for each data set is given, including point density and uncertainty in the height value. This is found by sampling each data set in the study areas and performing a profile analysis, using the software QT Modeler [27]. Figure 3.8 shows an example of this.

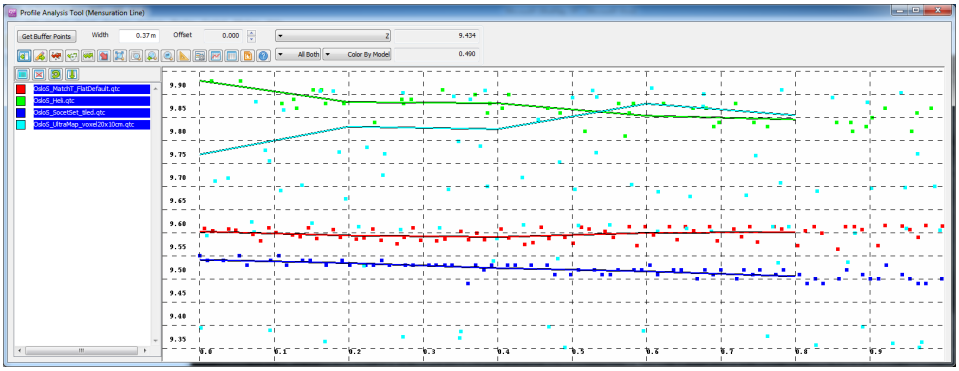


Figure 3.8: Profile analysis tool in QT Modeler. For this particular sample we notice the high Z-variability of UltraMap (turquoise, $\sim 50cm$) and the low Z-variability of Match-T (red, $\sim 5cm$).

This tool shows how much each point cloud's height values varies. In the example seen above, UltraMap's point cloud has about 50cm variation and Match-T has about 5cm. Ten such measurements were taken from the study areas and the mean value of these are considered the height uncertainty for each software.

3.5 Comparative analysis

To assess the results of the different software packages, the datasets were compared to data from a LIDAR scan of the same area. The imagery and the LIDAR data were captured a year apart. To verify the areas with respect to change in topography, road work and other changes that could affect the result, the historical photo feature of Opplysningen 1881's online map service [28] was consulted. The areas that were selected are described above, in Section 3.1.

3.5.1 Workflow

For each area, the following workflow was performed to compare the point clouds generated from the areal imagery to those of the LIDAR scans.

Using the ArcGIS software package [29], a polygon representing the area was registered and stored in a shapefile. This polygon was used as input along with the point clouds from each software, as well as the LIDAR data, in LasClip [30]. The output of LasClip is one single point cloud for each software and the LIDAR data, covering the area. From these point cloud files, gridded ($1 * 1$m) DSM rasters were created in QT Modeler. The DSM heights from the LIDAR dataset was lowered by 37cm to get the same height reference, as explained in Section 3.2.3.

Using the raster calculator in ArcGIS, a subtraction raster is created between the DSM from each point cloud and the one from the LIDAR dataset. These subtraction rasters form the basis of the comparative analysis. The statistics of the subtraction rasters were performed in MATLAB [31]. The script, included in the appendix, calculates the counts how many subtraction values each raster contains, the mean and standard deviation of these. These statistics are also calculated after removing gross errors, which in this case has been defined as having an absolute value > 1m.

Part III

Result

Chapter 4

Descriptive statistics and visual characteristics of point clouds

This section will give a description of each dataset visually and provide key statistical data for them. These include point density and noise rate. Differences resulting from different settings of each software package will also be presented. The results of the profile analysis is presented in Table 4.1. The LIDAR dataset is included as a reference.

Vertical uncertainty of point clouds [cm]					
Area sample #	LIDAR	Match-T	Socet Set	UltraMap	Description of surface
Oslo S 1	5	5	12	60	Roof
Oslo S 2	5	22	2	61	Asphalt road
Oslo S 3	8	4	4	55	Concrete platform
Oslo S 4	6	7	16	70	Grass
Botanisk Hage 1	16	30	18	11	Grass
Botanisk Hage 2	9	17	5	13	Gravel path
Botanisk Hage 3	4	4	8	12	Asphalt
Botanisk Hage 4	7	8	8	14	Grass
Urtegata 1	8	10	23	45	Roof
Urtegata 2	60	58	53	84	Roof
Mean	16.92	16.56	14.96	27.68	

Table 4.1: Result of profile analysis.

4.1 Introductory remarks

The resulting point clouds from the different softwares can be categorized by two different characteristics. The first has the points laid out like a ”blanket” covering the terrain and

buildings. In this version, there would typically be one point per XY-value and the noise rate would be low. The other kind is a more raw surface representation. From a distance, the result looks very pleasing, as objects that are flat will be represented as such. On close range, however, one can see that the surface representation is noisy, the height values have a high uncertainty.

4.2 Match-T DSM

Visual characteristics

The resulting point cloud from performing image matching in Match-T belongs to the blanket characteristic described above. The points are evenly distributed in a grid with one point for every $10 * 10$cm in the XY-plane. Each point's Z-value correlate with that of the neighbor. The noise rate is to be low, as can be seen in the profile comparison in Figure 4.1.

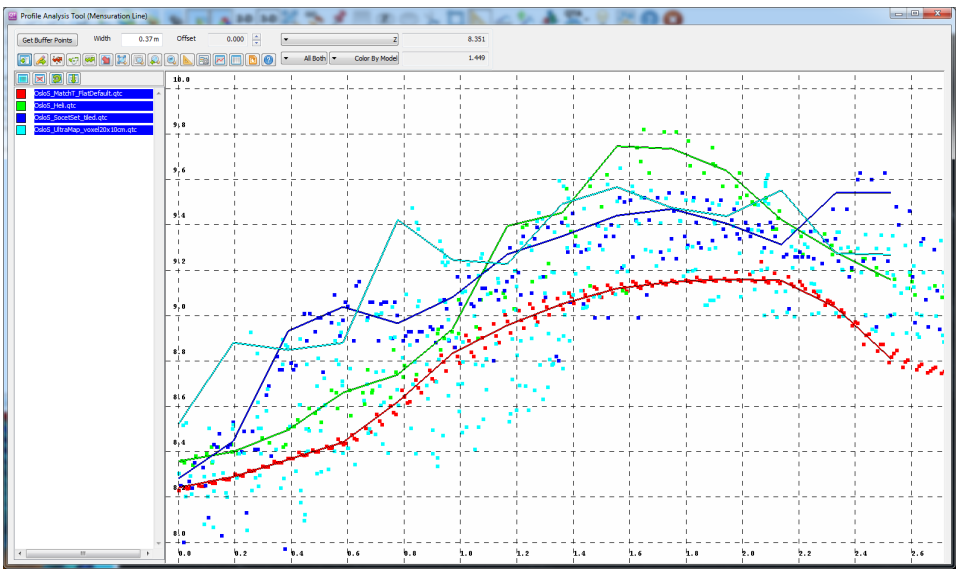
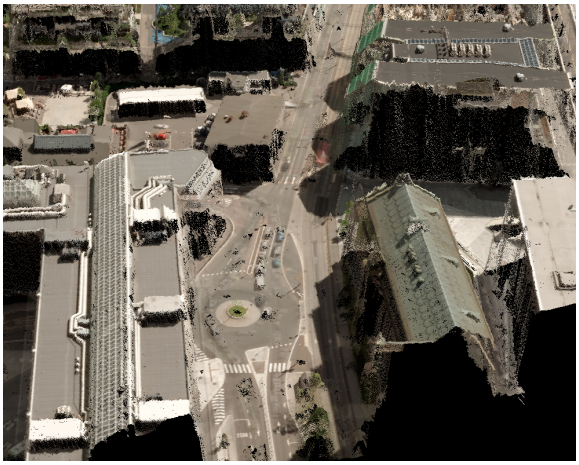
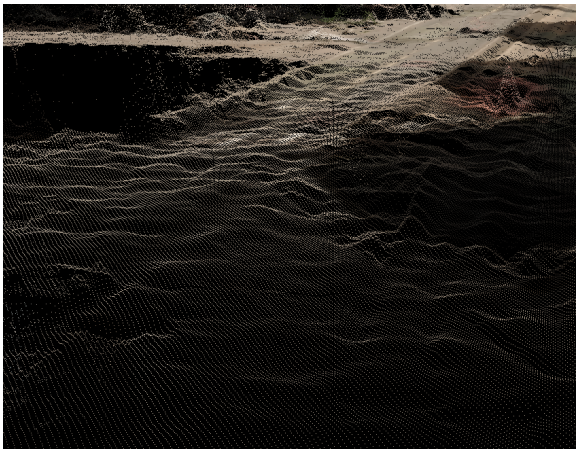


Figure 4.1: Profile analysis comparison of point clouds show the noise rate of each software (Red=Match-T, Blue=Socet Set, Turquoise=UltraMap) and LIDAR data (green).

These factors make the point cloud look neat and tidy. When zooming close to features certain artifacts arise. Specifically areas that should be flat, tend to undulate. Figure 4.2 shows a what the point cloud looks like from a distance (a), and the undulating tendency that becomes apparent when zooming in (b). Table 4.2 sums up key features of the resulting point cloud that was used in the comparative analysis in Chapter 5.



(a)



(b)

Figure 4.2: Point cloud from Match-T, the closer look in b) shows the undulating tendency on flat surfaces

Match-T DSM point cloud key features	
Visual characteristic	Blanket characteristic. Regular pattern of points. Neighboring points' height value correlates. Undulating terrain, even when representing flat surfaces.
Includes color	Yes, requires extra step through command line tool
Point density	$\sim 90 \frac{p}{m^2}$
Height uncertainty	$\sim$ 17cm

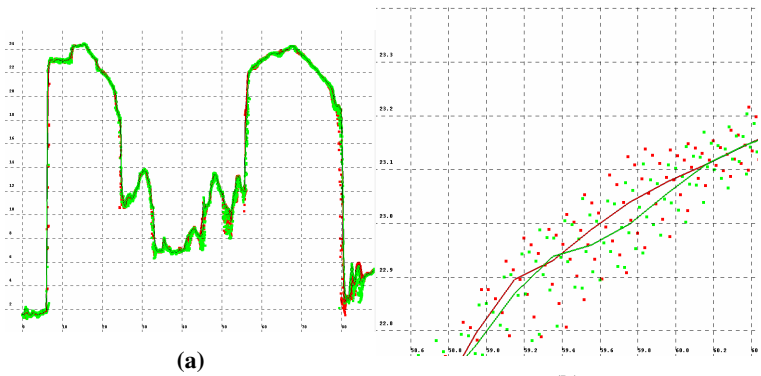
Table 4.2: Match-T DSM key features

Impacts of different settings

Match-T allows the user to set many parameters. Two different exports were performed in this study. The settings for the one used in this study are shown in Table 3.3. The alternative point cloud's settings are presented in the table below.

Match-T DSM, Alternative settings	
Generation type	DSM
Terrain type	Mountainous
Smoothing	Low
Feature density	High
Parallax threshold	18 pixels

Since the study area is urban, with lots of height variation and dense features, these settings were chosen. Very little difference between the resulting point clouds were found. Figure 4.3 shows a profile comparison of the two.



(a) (b)

Figure 4.3: Profile analysis of different parameter settings, Match-T. Green: Original, Red: Alternative settings.

4.3 Socet Set NGATE

Visual characteristics

What separates Socet Set from the other softwares is that (unless gridded is enabled in settings) it does not force points in areas with low correspondence. This means that areas where matching is difficult (surfaces with few distinct features) will have holes in them. This makes affects the visual appearance of the cloud to a degree, but for the most part the result looks pleasing. Flat surfaces are generally presented correctly and the noise rate is low. Table 4.3 sums up key features of the resulting point cloud that was used in the comparative analysis in Chapter 5.

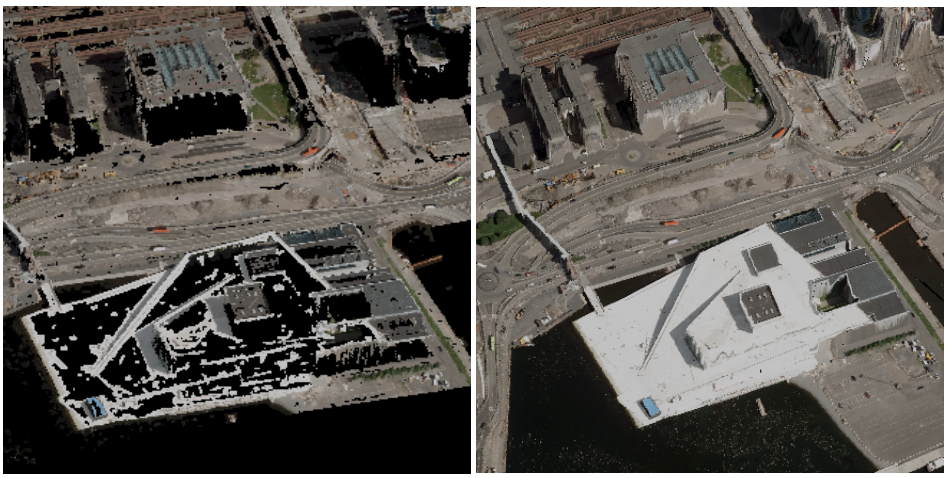
Socet Set NGATE point cloud key features	
Visual characteristic	Blanket characteristic. Regular pattern of points, but holes in areas that are difficult to match. Natural representation of terrain.
Includes color	No
Point density	$\sim 90 \frac{p}{m^2}$
Height uncertainty	$\sim$ 15cm

Table 4.3: Socet Set NGATE point cloud key features

Impacts of different settings

Socet Set NGATE allows for a several user inputs that may have an impact on the resulting point cloud. These are described in Section 3.3.2.

One of the most visually striking settings is gridded or ungridded output. If gridded is on, the software will force a point for every grid, which leads to a dense, regular point cloud. This can be very visually pleasing, but comparing the output to that of the same point cloud with this setting turned off using the same technique as in Chapter 5, showed slightly worse mean and standard deviation.



(a) Ungridded **(b)** Gridded

Figure 4.4: Ungridded vs. gridded point cloud from Socet Set. The gridded version insert points for each grid. The pointclouds are colored in the visualization software using orthophotos.

4.4 UltraMap

Visual characteristics

The first impression of the resulting point clouds from UltraMap is very good. Colors are included and represented accurately in the resulting LAS-file. The surface seems to be nearly completely covered, with only water and facades showing holes. The topology of the surface is accurately represented, with flat areas represented flat in the point cloud. The visual first impression of the result from UltraMap can be seen in Figure 4.5. On close range, however, it becomes apparent that there is a high variability in the height values of each point. The tendency is that, for every surface, the points are placed in a layer spanning 40-60cm, as seen in Figure 4.6.

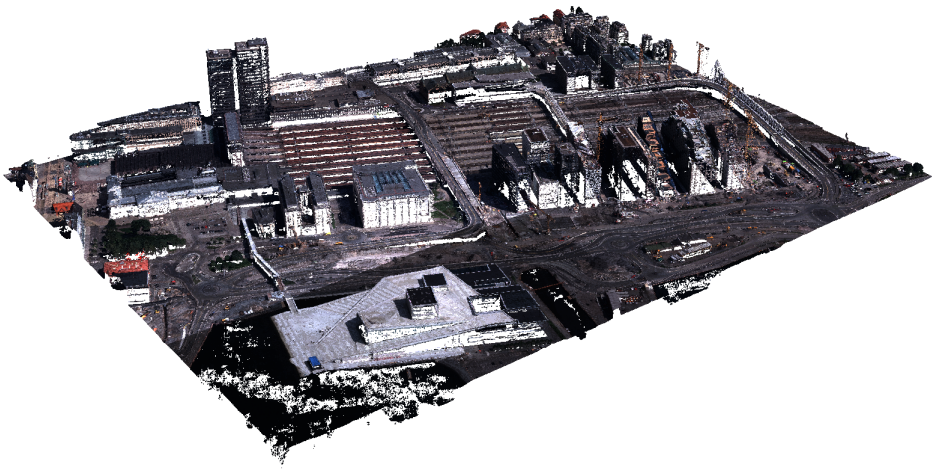


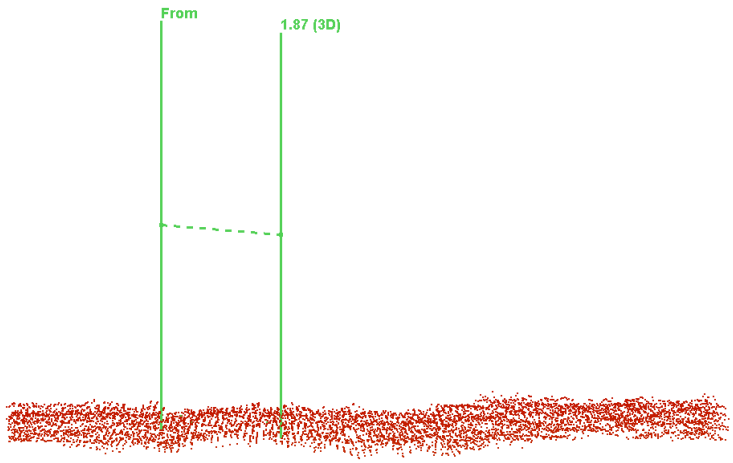
Figure 4.5: Visual impression of point cloud from UltraMap.

UltraMap point cloud key features	
Visual characteristic	Raw point cloud characteristic. Natural representation of surface, but noisy.
Includes color	Yes
Point density	$\sim 90 \frac{p}{m^2}$
Height uncertainty	$\sim$ 43cm

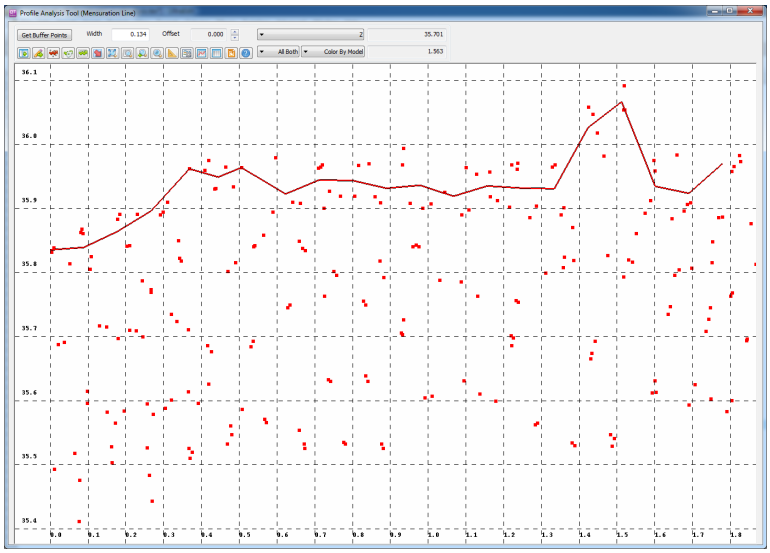
Table 4.4: UltraMap point cloud key features

Impacts of different settings

The settings for UltraMap are presented in Section 3.3.3. There are two possible parameter settings that directly impacts the point cloud. One is setting the DSM Processing Mode



(a) Cropped flat section of point cloud, seen from the side.



(b) Profile analysis.

Figure 4.6: UltraMap pointcloud height variability.

to standard in stead of robust. According to the reference manual [26], robust is supposed to be used if standard does not deliver satisfying results. Standard mode was tested in preliminary exports in this study. No visible differences were found in the results.

The other setting is horizontal and vertical cell size. The GSD resolution (10cm in the study) were kept for the vertical cell size. For the horizontal cell size, different values were tested, and except for the point cloud density, little difference with respect to visual characteristics and vertical uncertainty was found in the result.

Chapter 5

Comparative analysis

In this chapter, the data from the image matching is compared to that of laser data collected from a helicopter in 2011. For this analysis, DSM rasters were derived from each point cloud, both from the LIDAR and image matching. A subtraction raster between each image matching DSM and the LIDAR DSM were then calculated and analyzed. The workflow to achieve the subtraction datasets is described in Section 3.5.1.

5.1 Oslo S

Table 5.1 shows a summary of the DSM subtractions between each software and the helicopter data set. This area yields ~ 76000 difference values for Match-T and Socet Set, while Ultramap has a little less ($\sim 2\%$). Socet Set gets a higher SD on the raw data compared to the others (~ 8.3m vs. ~ 3.5m and ~ 3.1m). The mean values of the raw data varies among the three softwares, with UltraMap ending at a ~ -0.4m mean difference to the LIDAR data.

Removing gross errors improves these values for all softwares. In Socet Set, 29% of the difference values are removed, leaving 544660 values. Match-T and UltraMap retains 60103 and 61640, respectively. Interestingly, the mean value of Match-T improved more than the others, and is now at -0.04m compared to -0.114m for UltraMap and -0.116m for Socet Set. The standard deviation is similar for all three at 0.272m for UltraMap, 0.285m for Socet Set and 0.290m for Match-T.

5.2 Urtegata

In this area, all the softwares achieved 2604 subtraction values before removing gross errors, as seen in Table 5.2. As with the Oslo S area, Socet Set has a mean and standard deviation in the order of ~ 2 compared to that of UltraMap and Match-T.

Removing gross errors takes away 21% of the points from Socet Set and about 7% from Match-T and UltraMap. As in the previous area, Match-T shows a lower mean

DSM subtraction, Oslo S, all data values			
	Match-T	Socet Set	UltraMap
N	76369	76361	74751
Mean [m]	-0.722	-1.5356	-0.429
SD [m]	3.474	8.357	3.074
DSM subtraction, Oslo S, gross errors excluded			
	Match-T	Socet Set	UltraMap
N	60103	54660	61640
Mean [m]	-0.040	-0.116	-0.114
SD [m]	0.290	0.285	0.272

Table 5.1: Summary, DSM subtraction, Oslo S

difference than the others, while its SD is higher. UtraMap achieves a smaller SD at 0.186m compared to 0.277m and 0.250m for Match-T and Socet Set.

DSM subtraction, Urtegata, all data values			
	Match-T	Socet Set	UltraMap
N	2604	2604	2604
Mean [m]	-0.535	-1.278	-0.721
SD [m]	2.855	4.590	2.833
DSM subtraction, Urtegata, gross errors excluded			
	Match-T	Socet Set	UltraMap
N	2422	2070	2427
Mean [m]	-0.069	-0.263	-0.169
SD [m]	0.277	0.250	0.186

Table 5.2: Summary, DSM subtraction, Urtegata

5.3 Botanisk hage

As for the previous areas, Match-T and UltraMap achieves similar number of subtraction values before removing gross values, while, as in the Oslo S area, Socet Set has fewer. Table 5.3 shows that they have about 35800 values, compared to 29758 for Socet Set. The mean and standard deviations are closer here, than for the previous areas, but Socet set still has higher absolute values than the others.

Removing gross errors has a higher impact here for all the softwares, compared to Oslo S and Urtegata. For Botanisk Hage, the filtering takes away about half of the points for each software. After the filtering, they continue to have similar values. UltraMap achieves the lowest absolute mean value at -0.187m, while Match-T achieves -0.218m and Socet Set -0.232m.

DSM subtraction, Botanisk hage, all data values			
	Match-T	Socet Set	UltraMap
N	35864	29758	35861
Mean [m]	-1.944	-2.270	-2.037
SD [m]	4.845	5.213	4.956
DSM subtraction, Botanisk hage, gross errors excluded			
	Match-T	Socet Set	UltraMap
N	18253	14648	18349
Mean [m]	-0.218	-0.232	-0.187
SD [m]	0.345	0.322	0.331

Table 5.3: Summary, DSM subtraction, Botanisk hage

Part IV

Discussion and conclusions

Chapter 6

Discussion

6.1 Visual characteristics of the resulting photogrammetric point clouds

In Chapter 4, the results of each software were presented by its visual characteristics and descriptive statistics. The results showed that the resulting point clouds can be categorized by two different characteristics.

The blanket characteristic

The height value of each point in the cloud is dependent of the neighboring point's height value. This indicates that some sort of filtering algorithm has been applied to the point cloud, and that the result is intended to be close to a finished product. This effect can be visually pleasing if the surface is accurately represented by the point cloud. However, artifacts and unnatural representations of objects, like undulating surfaces on roads or flat buildings will be very noticeable in this type of representation.

The raw point cloud characteristic

Height values of points are not correlated to the neighboring ones. This indicates that the result is unfiltered, that the points comes directly from the image match, and that post-processing may be necessary for a more visually pleasing result.

6.2 Comparative analysis

The comparative analysis, presented in Chapter 5, showed differences in each software's accuracy. It should be noted that these results do include some caveats, as discussed in Section 3.5.

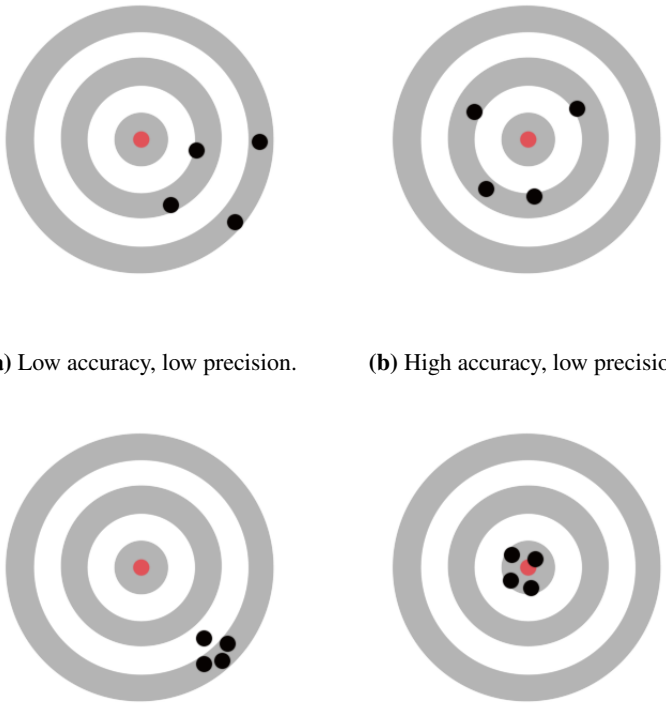
Accuracy and precision

The following sections will discuss some of the results with respect to accuracy and precision. In the context of the scientific method, the terms accuracy and precision are used to determine the validity of a measurement.

Accuracy: Indicates how close measurements of a system are to the true value.

Precision: Refers to the *repeatability* of a system's measurements.

A common way to explain accuracy and precision is the target analogy. In Figure 6.1, four targets and a variety of hits are shown.



(a) Low accuracy, low precision.

(b) High accuracy, low precision.

(c) Low accuracy, high precicion.

(d) High accuracy, high precision.

Figure 6.1: The target analogy is often used to explain the difference between accuracy and precision

Match-T

The most notable part of Match-T's results is that its mean difference value was remarkably good after removing gross errors. Its standard deviation is about the same as for

the other softwares. When considering its undulating characteristic, this is consistent of measurements with high accuracy, but lower precision.

The blanket characteristic of the point cloud from Match-T is an indication of post-matching filtering. Heuchel et al. [32] mentions a smoothing algorithm that were implemented between version 5.3 and 5.4 of the software. The algorithm consists of an adaptive filter that looks for discontinuities in the DSM. If the signal to noise ratio in an area is below a tolerance, the algorithm will smooth the surface with the surrounding points. The good results in the mean difference show that the algorithm proves efficient. The undulating characteristic, even for flat surfaces, and the higher standard deviation show that this algorithm may need improvement in recognizing flat surfaces.

Socet Set

As explained in Section 4.3 and seen in Figure 4.4 a), when format is set to TIN Triangles, Socet Set will not "force" a point unless it reaches a certain level of confidence. Despite this, removing gross errors in the comparative analysis removed more points for Socet Set than the others. This is surprising as, intuitively, Socet Set should already have excluded blunders.

For the areas with mostly urban features, Oslo S and Urtegata, the number of subtraction values were similar as for the other softwares. When removing gross errors (> 1m), significally more were filtered out for Socet Set. For the area with most vegetation, Botanisk Hage, the rejection rate was about the same for all softwares ($\sim 50\%$). However, Socet Set had already many fewer subtraction values than the others to start with.

UltraMap

Despite its high variability, the point cloud from UltraMap performed relatively stable, compared to the other softwares. Where Match-T showed best mean values, UltraMap showed similar improvements, though not as extreme as Match-T with respect to the mean. It did show the lowest standard deviation in most of the measurements. In the context of accuracy and precision, this indicates that the UltraMap system is the most precise system. At the same time, it does well with respect to accuracy.

6.3 Image matching vs. laser data

The software manufacturers present the application of dense image matching for photogrammetric purposes as a possible alternative to traditional airborne laser scanning. This study has been a comparison between different image matching softwares, not between image matching and LIDAR. Still, the comparative analysis has shown that the LIDAR data and matched image data can be quantitatively compared. They still have certain qualitative differences, that may be of importance to different use cases.

If point cloud data of an area is the main objective of a project, a LIDAR survey still has advantages over aerial imagery. The processing of the image data may require days, including import, aerotriangulation, and radiometry adjustments before the image

matching can begin. The image matching itself is resource-intensive. For the area in this study, which is relatively limited in size, the matching process of the different softwares took between 10 and 20 hours to complete. With LIDAR data, the data is already in point cloud format and ready for further analysis.

The LIDAR beams penetrates vegetation and the LAS format [33] has the capability of storing multiple returns of each laser beam. Usually the first return represents the surface, including treetops. The last return typically represents the ground points in areas covered by vegetation. This can be useful in forestry analysis and gives the possibility to process a DTM (Digital Terrain Model) from the data in addition to a DSM (Digital Surface Model). Photogrammetric point clouds obviously does not have this capability, as the image sensor only has one "return".

For projects that already need aerial imagery, e.g. for orthophoto production, these new applications can provide further value to the already captured data at little extra cost.

6.4 Effect of aerotriangulation

Shedding light on the effect of the aerotriangulation (AT) is a part of the problem description of this study. As AT is a standard process within the image processing workflow (Section 3.2.1), this and bundle adjustment was performed within the UltraMap framework. The nature of the image processing workflow and the time constraints of this study meant that the AT was performed once and not revisited. The result of the AT was exported to a BINGO project. All three softwares in this study supports BINGO, so this ensured that they could be compared on equal terms.

A potential source of error in the AT result is that only two ground control points (GCP) were available in the study area. The GCPs, seen in Figure 6.2, were provided by Oslo Municipality and there are no more points available within the study area.

Ideally, there should be at least one more point in order to complete the absolute orientation [34]. The lack of a third point can lead to height errors, the effect of which is higher further away from the GCPs. To amend this, surveying more GCPs could have been done. Due to time constraints, and since the analysis were between the different softwares, and they all had equal basis, this was not done.

Figure 6.2: Ground Control Points in the study area, marked in yellow.

Chapter 7

Conclusions and further work

7.1 Conclusions

This study consisted of studying and comparing different algorithms and software packages that perform image matching on aerial photography to generate point clouds. The results showed that the point clouds from modern image matching software are comparable to those of a LIDAR survey.

The comparative analysis revealed that the point cloud from Socet Set NGATE, despite discarding uncertain height values in its matching process, contained more gross errors than the others. UltraMap's point cloud performed most stable of the three softwares, with decent (compared to the others) mean difference and standard deviation value in all the study areas. Match-T DSM performed well in the urban areas (Oslo S and Urtegata), and after filtering out gross errors in the subtraction datasets, it had a significantly better mean value than the others. The standard deviation was still about the same as the others, indicating a high accuracy and medium precision.

With regards to the visual characteristics of the point clouds, it was found that Socet Set and Match-T have similar traits. Both of these softwares perform a filtering algorithm when generating the point clouds, making the point clouds visually pleasing with a low noise-rate. The points are placed over the terrain like a blanket, and neighboring point's height values influence each other. This can lead to certain artifacts, like difficulty in representing flat terrain. UltraMap does not perform this filtering algorithm on export, and its point cloud is a more raw representation of the image matching process. This leads to more noise in the output, and more post-processing may be required for a more pleasing visual result.

7.2 Further work

Absolute comparison of photogrammetric point clouds

In this study, the point clouds from different softwares were compared, using LIDAR data as reference. For the purpose of this study, this was convenient, but ideally more absolute reference points would have been preferable. The results in Chapter 4 showed that the output of the image matching softwares were comparable to that of the LIDAR scan. Comparing to absolute reference points or planes would allow for a comparison of quality between LIDAR and image matching.

Closer analyses of each software package

Due to time constraints and time spent learning each software in this study, a close analysis of all the parameter settings was difficult. UltraMap allows for only a few settings to tweak, but Socet Set and Match-T allows the user to decide more of the parameters. The results of the comparative analysis in this study showed that the terrain type affected Match-T and Socet Set's performance significantly. Finding exactly the right settings for optimal results in different types of areas could be an interesting topic of further investigation.

Developing prototypes for the algorithms

Chapter 2 presents different algorithms for image matching. There are few open source alternatives to the software packages in this study, but some are starting to show up [35, 36, 37, 38]. Implementing prototypes of the different algorithms and testing their performance would be an interesting further topic of research.

Point cloud filtering techniques

UltraMap's output consisted of the raw height data from image matching process. Match-T had problems with representing the flat areas on the surface. Another research topic could be to explore filtering techniques to accurately display the urban surface.

References

[1] Norbert Haala. Comeback of digital image matching. *Photogrammetric Week*, 2009.

[2] Armin Gruen. Development and Status of Image Matching in Photogrammetry. *The Photogrammetric Record*, 27(137):36–57, March 2012. ISSN 0031868X.

[3] Jacqueline Moigne. *Image registration for remote sensing*. Cambridge University Press, Cambridge New York, 2011. ISBN 0521516110.

[4] Chris Glasbey and Graham Horgan. *Image analysis for the biological sciences*. John Wiley & Sons, Inc., New York, NY, USA, 1995. ISBN 0-471-93726-6.

[5] ESRI. Convolution function. URL `http://resources.arcgis.com/en/help/main/10.1/index.html#//009t0000004s000000`. [Cited: April 15, 2013].

[6] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. ISSN 01628828.

[7] MathWorks. Find edges in grayscale images. URL `http://www.mathworks.se/help/images/ref/edge.html`. [Cited: May 20, 2013]].

[8] Heiko Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):328–341, March 2008. ISSN 0162-8828.

[9] Barbara Zitova. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003. ISBN 076952334X. ISSN 02628856.

[10] Helmut Mayer, Monika Sester, and George Vosselman. *Basic computer vision techniques*. American Society for Photogrammetry and Remote Sensing, Bethesda, Md, 5 edition, 2004. ISBN 1570830711. 95–162 pp.

[11] Abdalla Alobeid, Karsten Jacobsen, and Christian Heipke. Comparison of Matching Algorithms for DSM Generation in Urban Areas from Ikonos Imagery. *Photogrammetric engineering and remote sensing*, 76(9):1041–1050. ISSN 0099-1112.

[12] Wolfgang Förstner. A feature based correspondence algorithm for image matching. *Archives of Photogrammetry and Remote Sensing*, 26(3):150–166, 1986.

[13] Toni Schenk. *Digital Photogrammetry*. TerraScience, 1999. URL http://books.google.no/books?id=O-RzSgAACAAJ.

[14] Geometric Computer Services. Bingo-f. URL http://www.gcs-abc.com/bingof.htm. [Cited May 10, 2013].

[15] Kartverket. Transformasjoner. URL http://www.statkart.no/Posisjonstjenester/Transformasjoner/. [Cited: June 4, 2013].

[16] Kartverket. Norges offisielle høydesystemer og referansenivåer, 2009. URL http://www.kartverket.no/Documents/Standard/BransjestandarderutoverSOSI/Hoydesystemer_referansenivaaer.pdf.

[17] Inpho. Match-t dsm. URL http://www.inpho.de/index.php?seite=index_match-t. [Cited May 2, 2013].

[18] Trimble. *Reference Manual - Match-t dsm 5.5*. Trimble, 2012.

[19] BAE Systems. Next-generation automatic terrain extraction (ngate), white paper, 2012. URL http://www.socetgxp.com/docs/education/white_papers/wp_ngate.pdf. [Cited May 2, 2013].

[20] Kurt DeVenecia, Stewart Walker, and Bingcai Zhang. New approaches to generating and processing high resolution elevation data with imagery. *Photogrammetric Week*, (4), 2007.

[21] Microsoft. Ultramap 3.0. URL http://www.microsoft.com/ultracam/en-us/UltraMap.aspx. [Cited May 20].

[22] Alexander Wiechert, Michael Gruber, and Konrad Karner. UltraMap : The all in one photogrammetric solution. XXXIX(September):2010–2013, 2012.

[23] Bernard Reitinger, Mario Sormann, Zebedin, Bernhard Schachinger, and Hoefler. Ultramap V3 – A Revolution In Aerial Photogrammetry. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 39(September):149–152, 2012.

[24] Heiko Hirschmüller. Semi-Global Matching Motivation, Developments and Applications. *Photogrammetric Week*, (Figure 1):173–184, 2011.

[25] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-Based Stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity Measure. *18th International Conference on Pattern Recognition (2006)*, 3(c):15–18, 2006. ISBN 0769525210. ISSN 10514651. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1699458.

[26] Vexcel Imaging GmbH. UltraMap 3.0 Ortho Production User Guide, 2013.

[27] Applied Imagery. Quick terrain modeler. URL http://www.appliedimagery.com/. [Cited: June 5, 2013].

[28] Opplysningen 1881. Kart på nett. URL http://www.1881.no/Kart/. [Cited: May 25, 2013].

[29] ESRI. Arcgis - mapping and spatial analysis for understanding our world. URL http://www.esri.com/software/arcgis. [Cited: May 25, 2013].

[30] Martin Isenburg. Lastools. URL http://www.cs.unc.edu/~isenburg/lastools/. [Cited: May 25, 2013].

[31] Mathworks. Matlab, the language of technical computing. URL http://www.mathworks.se/products/matlab/. [Cited: June 5, 2013].

[32] Tobias Heuchel and Andre Köstli. Towards a Next Level of Quality DSM/DTM Extraction with MATCH-T. *Photogrammetrische* . . . , pages 197–202, 2011.

[33] ASPRS. LAS Specification Verion 1.3. 2010. URL http://asprs.org/a/society/committees/standards/LAS_1_3_r11.pdf.

[34] Karl Kraus. *Photogrammetry : Geometry from Images and Laser Scans*. Walter De Gruyter, Berlin New York, 2007. ISBN 978-3-11-019007-6.

[35] Changchang Wu, S. Agarwal, B. Curless, and S.M. Seitz. Multicore bundle adjustment. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3057–3064, 2011. ISSN 1063-6919.

[36] Google. Ceres solver. URL https://code.google.com/p/ceres-solver/. [Cited June 1, 2013].

[37] Manolis Lourakis. sba : A generic sparse bundle adjustment c/c++ package based on the levenberg-marquardt algorithm. URL http://www.ics.forth.gr/~lourakis/sba/. [Cited June 1, 2013].

[38] Christopher Zach. Structure-and-motion (sfm) software. URL http://www.inf.ethz.ch/personal/chzach/opensource.html. [Cited June 1, 2013].

Appendix

The following MATLAB scripts for low-pass, high-pass and edge-detection filters were used to create the images in Figure 2.2, 2.3, and 2.4, respectively.

Low-pass filter

```
clear all;

I = imread('spektrum_crop.jpg');
smoothKernel = [1,1,1;1,1,1;1,1,1]/9;
imSmooth = imfilter(I,smoothKernel);

figure
subplot(1,2,1);
imshow(I); title('Input image');

subplot(1,2,2);
imshow(imSmooth); title('Smoothed image');
```

Sharpening filter

```
function [sharpImage] = highpass(image)
%Function takes a graphics file as input
%and returns a sharpened version

%Define sharpening filter:
highPassKernel = [-1,-1,-1;-1,9,-1;-1,-1,-1]/8;
%Apply filter to image:
highPassImage = imfilter(image, highPassKernel);

%adding the high pass image to the original yields a sharpened image
sharpImage = image + highPassImage;
```

Edge detection filter

```
function [imEdge, smoothImEdge] = edgedet(image)

%circular, laplacian edge detection filter:
edgeKernel = [0,-1,0;-1,4,-1;0,-1,0];

%Only edge detection:
imEdge = imfilter(image, edgeKernel);


%First blur, then edge detection
smoothKernel = [1,1,1;1,1,1;1,1,1]/9;
smoothImage = imfilter(image,smoothKernel);
smoothImEdge = imfilter(smoothImage, edgeKernel);
```

Raster statistics

MATLAB script for calculating raster statistics, used on the subtraction datasets in the comparative analysis in Chapter 5.

```
function [meanA,meanB,sdA,sdB] = meanImage (image)
%function takes subtraction raster as input and calculates ...
    statistics on it

num=0;
sum=0;

hits=0;
sum2=0;

a = [];
b = [];

largerthan05 = 0;

for n=1:size(image,1)
    for m=1:size(image,2)
        if(image(n,m)>-100)%nodata values
            num=num+1;
            sum=sum+image(n,m);
            a=[a image(n,m)];
        end

        if(abs(image(n,m))<1)%gross errors
            hits=hits+1;
            sum2=sum2+image(n,m);
            b=[b image(n,m)];
            if(abs(image(n,m))>0.5)
                largerthan05=largerthan05+1;
            end
        end
    end
end

sdA = std(a);
minA = min(a);
maxA = max(a);
meanA = mean(a);

sdB = std(b);
minB = min(b);
maxB = max(b);
meanB = mean(b);
```