

GUIDED CLOSE RANGE PHOTOGRAMMETRY FOR 3D MODELLING OF CULTURAL HERITAGE SITES

Bashar Saleem Abbas Alsadik

Examining committee:

Prof.dr.ir. A. Stein
Dr. A.A. Voinov
Prof.dr.-Ing. P. Grussenmeyer
Prof.dr. J. Mills

University of Twente
University of Twente
INSA Strasbourg
Newcastle University

ITC dissertation number 260
ITC, P.O. Box 217, 7500 AE Enschede, The Netherlands

ISBN 978-90-365-3793-3
DOI 10.3990/1.9789036537933
Cover designed by Job Duim
Printed by ITC Printing Department
Copyright © 2014 by B.S.A. Alsadik



UNIVERSITY OF TWENTE.

ITC

FACULTY OF GEO-INFORMATION SCIENCE AND EARTH OBSERVATION

GUIDED CLOSE RANGE PHOTOGRAMMETRY FOR 3D MODELLING OF CULTURAL HERITAGE SITES

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof.dr. H. Brinksma,
on account of the decision of the graduation committee,
to be publicly defended
on Friday 21st November 2014 at 12:45 hrs

by

Bashar Saleem Abbas Alsadik

born on July 01,1970

in Babylon, Iraq

This thesis is approved by
Prof.dr.ir. M.G. Vosselman, promoter
Dr. M. Gerke, co-promoter

To my family

Acknowledgements

After a long time of waiting, suffer, and hard work I am reaching my dream to have the PhD. degree. This was not possible to become a truth without the contribution of many people in my life who deserve to be acknowledged here.

I gratefully acknowledge my promoter Prof. George Vosselman for giving me the opportunity to join his group TOPMAP and to be one of his students. During the four years of research, he was very noble, positive, and understanding.

My deep thanks to my daily supervisor Dr. Markus Gerke. The research journey started in 2010 with him and he put a great effort and help to figure out together the possible research topic. Later, during the research years, Markus was a great supervisor and friend.

My sincere acknowledgment to the University Assistance Fund UAF and Scholars At Risk SAR. They gave me a great support during my stay in the Netherlands.

My gratitude to the University of Twente UT for offering me the position of promovendus (Aio) during the four years of my research.

I introduce my acknowledgment to ITC - EOS department staff members, especially Teresa Brefeld who showed a great help and support during my stay and Mila Koeva who kindly designed the thesis cover.

I would also like to introduce my thanks to all my colleagues in ITC and particularly Yaseen Taha (Iraq), Juan Pablo (Colombia), Jing Xiao (China), Ahmed A. Ibrahim (Egypt), Biao Xiong (China), Sudan Xu (China), MengMeng Li (China), Adugna Mullissa (Ethiopia), Anand Vetrivel (India), Dimo Todorovski (Macedonia).

I extend my thanks to the members of 3DOM – FBK in Italy, namely, Fabio Remondino, Francesco Nex, Fabio Menna, and Erica Nocerino.

Importantly, I can never express my deepest gratitude to my wife, sons, parents, brothers and sisters for their limitless love and support over the years.

This thesis is dedicated to my family.

Table of contents

Acknowledgements	i
Table of contents	iii
List of figures	vi
List of tables.....	xii
Chapter 1 Introduction	1
1.1 Preface	1
1.2 Research problem	6
1.3 Research objectives	9
1.4 Scope and limitations	11
1.5 Innovations in this work	11
1.6 Thesis Outline	12
Chapter 2 Literature Review	13
2.1 Design (sensor and network geometry)	13
2.2 Three Dimensional Measurements And Reconstruction.....	15
2.2.1 Structure from Motion (SfM)	15
2.2.2 Dense Matching	16
2.2.3 Modelling and Texturing	17
2.3 State-of-the-art software in reality IBM	18
2.4 Shortcoming in the IBM State-of-the-art.....	20
2.5 Visibility.....	21
2.5.1 Surface triangulation based methods	23
2.5.2 Voxel based approach	25
2.5.3 Hidden point removal (HPR).....	26
Chapter 3 Overview of Methodology.....	29
3.1 Camera network design	30
3.2 Guiding the image capture	30
3.3 Guiding the image tie points matching.....	31
3.4 Detection and recovery of gaps in the 3D point cloud	32
Chapter 4 Camera Network Design	35
4.1 Initial point cloud	36
4.2 The subdivision of the point cloud and camera domain	37
4.2.1 The identification of imaging location	39
4.2.2 The camera domain placement.....	40
4.2.3 The subdivision into facades.....	41
4.3 Initial camera network design in blocks	44
4.4 Connecting cameras.....	47
4.5 Minimal camera network.....	48
4.5.1 A minimal camera network based on filtering for coverage...	49

4.5.2	A minimal camera network based on the accuracy of the object points	50
4.6	Visibility requirement	52
4.7	Optimization technique	52
4.7.1	The formulation of the camera network optimization problem	54
4.7.2	The constraints on optimum camera placement	55
4.8	Summary of methodology	57
4.9	Experimental tests	58
4.9.1	Simulation test	59
4.9.2	Building	63
4.9.3	Statue	68
4.9.4	Discussion of results	71
4.10	Summary	73
Chapter 5	Guided Image Capture and Post Processing	75
5.1	Guiding the image capture	76
5.1.1	Creating the synthetic images	77
5.1.2	Real image capture	80
5.1.3	Matching between real and synthetic images	81
5.1.4	The validation of the real captured images	83
5.2	Post processing computations	86
5.2.1	Image connectivity for matching	86
5.2.2	Guided image matching by exploiting the object model	88
5.3	Experiment	91
5.4	Summary	95
Chapter 6	Gap Detection and Final Modelling	99
6.1	Gaps possible causes	99
6.1.1	The insufficient object coverage by images	100
6.1.2	The complexity of the object itself	100
6.1.3	The occluding objects between the camera and the object	101
6.1.4	The poorly textured parts of the object	102
6.1.5	The existence of real holes (openings) in the object	102
6.2	Gaps detection techniques	105
6.2.1	2D-based processing	105
6.2.2	Gap detection by voxelization	111
6.3	The auxiliary image capture and orientation	122
6.4	Summary	125
Chapter 7	Case Study	127
7.1	Referencing data	127
7.2	Camera network design	128
7.3	Guiding the image capture	131
7.4	Extraction of image correspondences	132

7.5	Image orientation	135
7.6	Gap detection and final modelling	140
7.7	Summary	147
Chapter 8	Conclusions and recommendations	149
8.1	Conclusions.....	149
8.2	Recommendations.....	151
Appendix A.....		153
A1-	Optimization.....	153
A2-	Cost function in optimal camera network design	155
Bibliography		157
List of publications		164
Summary		165
Samenvatting		167
Curriculum Vitae		169
ITC Dissertation List		170

List of figures

1.1	Imaging geometry. (a) Small B/D ratio and large error. (b) Large B/D ratio and small error	2
1.2	Automated 3D model from (37) still shot images (left) and 3D model from (178) video frames (right)	3
1.3	Repetitive texture patterns affecting the tie points matching and leading to wrong correspondences	4
1.4	Failure examples of SfM	5
1.5	Façade point cloud from the short baseline (left) and façade point cloud from the wide baseline (right) using Agisoft.....	6
1.6	3D modelling of cultural heritage by imaging techniques	9
1.7	The summary of research objectives and questions.....	11
2.1	General techniques and state-of-the-art software for IBM.....	20
2.2	Point cloud in an unknown looking position either forward or backward.....	21
2.3	Visibility by using the triangular surface normal vectors.....	23
2.4	Visibility by testing the ray-triangle intersection	24
2.5	Depth - buffering method	25
2.6	Voxel-ray intersection for visibility	25
2.7	HPR method (Katz et al. 2007)	27
3.1	The general methodology of the proposed camera network planning and guidance	29
3.2	The methodology of camera network design	30
3.3	General methodology of the proposed guiding system.....	32
3.4	The gap detection methodology.....	33
4.1	The general methodology of camera network design	36
4.2	Initial point clouds and video track.....	37
4.3	(a) Subdivide the building point cloud into its facades. (b) Two types of camera networks for a U-shaped building with the subdivision into facades and the inefficient ring network design.	38
4.4	Workflow of finding the camera domain of a rough cloud of a building.....	39
4.5	Identification of cameras to be placed inside or outside the building.(a) Inward camera domain ($R>r$). (b) Outward camera domain ($R<r$).....	39
4.6	The angular orientation of suitable camera domains for. (a) Outward camera domain object. (b) Inward camera domain	40
4.7	The camera circular domain for statues and monuments	40
4.8	Point cloud subdivision methodology	41
4.9	Edges extraction by testing the normal.....	42
4.10	Point cloud subdivision technique.....	43

4.11	Surface normals are used to avoid interference of sub-clouds	43
4.12	Photo layout for standard ordered-block in terrestrial photogrammetry	44
4.13	The relation between the camera accessible heights and object height	45
4.14	Camera and object coordinate systems	47
4.15	Connecting camera placement around a building and its importance for 3D modelling	48
4.16	The minimal camera network in the general workflow of design	49
4.17	The concept of filtering redundant camera. (a) Before filtering. (b) After filtering. (c) Number of covering cameras before and after the filtering	49
4.18	The workflow diagram of filtering for coverage or accuracy requirements	51
4.19	The effect of visibility analysis on the design. (a) A simple building framework with normals. (b) The dense network configuration. (c) Minimal network with visibility test. (d) Minimal network without visibility test.....	52
4.20	The workflow of the optimization computations	53
4.21	The bounding limits of the optimal camera orientation	56
4.22	The general workflow of the camera network design	58
4.23	A simulated dense camera network.....	59
4.24	(a) The point precision before optimization. (b) The point precision after optimization. (c) The convergence of solution of optimization. (d) The number of the covering cameras before and after the optimization.....	60
4.25	The simulation of the filtered imaging network and results of filtering and optimization. (a) Filtered net for coverage. (b) Optimal net. (c) Optimization run plot. (d) The number of the covering cameras before and after the optimization	61
4.26	(a) The error after the filtering for accuracy. (b) The error after the optimization. (c) The convergence of solution of optimization. (d) The frequency of covering cameras before and after the optimization.....	62
4.27	Video stream for a building (sample images).....	63
4.28	(a) The video imaging track around the building. (b) The derived rough model and mesh. (c) The image morphological operation before filtering. (d) Subdivision result	63
4.29	The building initial dense imaging block before filtration	64
4.30	Imaging network configuration of the building. (a) Filtered network for coverage. (b) Filtered network for accuracy. (c) Optimal network.....	65

4.31	The exaggerated error ellipsoid plot for the building experiment. (a) Dense network. (b) Filtered network for coverage. (c) Filtered network for accuracy. (d) Optimal network.....	66
4.32	Log plot of optimization runs for each detected façade of the building.....	67
4.33	The number of the viewing cameras per point. (a) The coverage in the dense network. (b) The coverage of the filtered network. (c) The coverage of the optimal network	67
4.34	(a) Statue surface mesh. (b) The dense imaging block	69
4.35	Statue imaging network. (a) Minimal network before optimization. (b) After optimization.....	69
4.36	The error ellipsoids for the statue points in three cases. (a) Dense network. (b) Filtered network. (c) Optimal network.....	70
4.37	Log plot of optimization runs of the statue imaging	71
4.38	The number of viewing cameras for each object point. (a) Dense network. (b) Filtered network. (c) Optimized network	71
5.1	Workflow of the guiding imaging system.....	77
5.2	Low-detailed 3D textured model.....	78
5.3	Texture transfer of synthetic images	79
5.4	Removing occluded pixel values by distance buffering	79
5.5	Synthetic image creation with and without self-occlusion computation.....	80
5.6	Software-hardware illustration of the guiding system	81
5.7	Restriction of the matching space. Left: SIFT points in synthetic image. Right: Restricted SIFT points in real image.....	82
5.8	SIFT matching between real and synthetic images.....	82
5.9	The refined points matching between synthetic and real image .	83
5.10	Space coordinates interpolation of the matching points. (a) The surface mesh projected on the synthetic image. (b) The coordinates interpolation	84
5.11	The space resection of the real-captured image.....	85
5.12	The validation of the real captured image workflow.....	85
5.13	The guiding system and validation	86
5.14	a) The image connectivity graph. (b) The matching matrix. (c) The pre-planned network of 106 images.....	87
5.15	Mismatching in the full pairwise approach leads to unsuccessful orientation. (a) The image pair. (b) Full pairwise approach (VSfM). (c) Guided approach (VSfM)	88
5.16	Keypoint estimation in high resolution images.....	89
5.17	The guided matching methodology by exploiting the 3D model..	90
5.18	The camera network design of a monument. (a) Dense network. (b) Filtered network from two views.....	91
5.19	(a) The synthetic images of the monument. (b) The real captured images.....	92

5.20	The oriented monument images shown in the web service of Photosynth	92
5.21	The histogram chart of the processing time for the full pairwise and the guided matching	93
5.22	Four-image sample set.....	93
5.23	Comparison of error ellipsoids. (a) SIFT matching. (b) SURF matching. (c) NCC matching	94
5.24	(a) Guided matching results in 15 oriented images. (b) Smart match (Photomodeler) results in 14 oriented images.....	95
5.25	(a) No. of oriented images. (b) No. of tie points. (c) The average point marking residuals [pixels]. (d) Average point precisions [m].....	95
6.1	Obscured areas from street level view.....	100
6.2	Gaps caused by object complexity a) Gaps due to self-occlusions. (b) Gaps due to protrusions.....	101
6.3	Gaps due to occlusions.....	102
6.4	Gaps due to textureless objects.....	102
6.5	Gaps due to openings.....	103
6.6	The effect of occlusions on the completeness of the final 3d point cloud.....	103
6.7	The effect of textureless parts, openings and self occlusion on the completeness of the final 3D point cloud	104
6.8	The effect of protrusions on the completeness of the final 3d point cloud.....	105
6.9	Triangulated surface mesh for gap detection. (a) Building with openings. (b) Facade with occluded gaps.....	106
6.10	Image morphology on the raster image of the projected point cloud. (a) Original image. (b) BW image of the PC. (c) Image closing with disk SE. (d) Image closing with line SE	107
6.11	Methodology of 2D-based gap detection	108
6.12	The 2D-based gap detection experiment. (a) Projecting the point cloud to the images. (b) Gap detection technique. (c) The 3d projected gap points.....	109
6.13	Façade dense point cloud.....	110
6.14	The gap detection in a building stairs	110
6.15	The workflow of the 3D based gap detection	111
6.16	The 3D voxelization of point cloud with 20 cm voxel side. (a) Point cloud. (b) Occupied voxels. (c) Empty voxels. (d) Combined visualization of the empty and occupied voxels.....	112
6.17	The initial filtering of voxels. (a) Before refinement (b) After refinement.....	112
6.18	Labelling empty voxels by visibility analysis. (a) Fully occluded. (b) Real opening. (c) Fully visible empty voxel	113
6.19	Labelling partly occluded empty voxels by visibility analysis. (a) Obstacle-occlusion effect. (b) Insufficient covering cameras or self-occlusion	113
6.20	(a) Proximity measure. (b) 3×3 neighbouring voxels.....	115

6.21	The output results of gap detection. (a) Occupied voxels. (b) Empty voxels labelled according to visibility. (c) Empty voxels After visibility test. (d) Final filtered empty occluded voxels	117
6.22	Statue dense point cloud and gaps	118
6.23	Gap detection in a statue point cloud. (a) Occupied voxels. (b) Empty voxels. (c) Empty voxels After visibility test	118
6.24	Gap detection in a statue point cloud	119
6.25	Gap detection of a façade with opening. (a) Point cloud. (b) Occupied voxels. (c) Empty voxels. (d) Empty and occupied voxels. (e) Visibility labelling. (f) Partly occluded voxels. (g) Empty voxels after neighbourhood analysis and filtering.....	120
6.26	Facade with real opening and visibility analysis result. (a) Point cloud. (b) Occupied voxels. (c) Empty voxels. (d) Empty and occupied voxels. (e) Visibility labelling. (f) Partly occluded voxels. (g) Final detected gaps.....	121
6.27	Image connectivity with auxiliary images.....	123
6.28	The dense point cloud of façade. (a) Detected gap voxels. (b) Dense point cloud before detection (b) dense point cloud after the auxiliary image capture.....	124
6.29	The dense point cloud of the statue base. (a) The detected gap voxels. (b) The dense point cloud before recovery. (c) The dense point cloud after the auxiliary image capture shown in (d)	125
7.1	The reference point cloud by using the terrestrial laser scanning	128
7.2	(a) The GCPs distribution. (b) The GCP target design	128
7.3	Accuracy estimation for the imaging design	129
7.4	The rough point cloud of the church using VSfM	129
7.5	The segmentation of the church rough point cloud	130
7.6	The camera planning and optimization of the church. (a) Dense network. (b) Filtered network. (c) Optimal network.....	131
7.7	Sample of synthetic and real captured images. (a) Synthetic images. (b) Real images. (c) SIFT matching between the synthetic and captured image. (d) Space resection	132
7.8	Captured data set for the 3D modelling of the church.....	132
7.9	Comparison between full pairwise and guided connectivity graph. (a) Connectivity graph of the Pre-designed network (1319 matches). (b) Full pairwise connectivity graph (6903 matches)	133
7.10	The time consumed in full pairwise and guided matching for the church dataset	133
7.11	Mismatching case of a repetitive pattern of the church left and right facades. (a) SIFT mismatching. (b) Guided approach. (c) Full pairwise approach.....	134
7.12	(a) Image correspondences using our guided matching. (b) Image correspondences using VSfM software. (c) Image correspondences using Agisoft photoscan software	135
7.13	The oriented camera network by guided image correspondences using APERO	136
7.14	The dense textured point cloud of the church using MICMAC	137

7.15	The first test for point cloud validation. (a) Image-based point cloud. (b) TLS point cloud. (c) Distance between the point clouds	138
7.16	The second test for point cloud validation. (a) Image-based point cloud. (b) TLS point cloud. (c) Distance between the point clouds	139
7.17	The validation of the upper part of the church point cloud.....	140
7.18	Gap detection with voxels. (a) The dense point cloud and occupied voxels. (b) The empty voxels are coloured according to visibility test. (c) Filtered voxels after visibility test and the final detected gaps	141
7.19	The church point cloud with potential detected gaps represented as red voxels	142
7.20	The captured auxiliary images in yellow.....	143
7.21	Dense point cloud illustrates the first detected gap in a column. (a) Before gap detection. (b) After adding the auxiliary images and modelling	144
7.22	Dense point cloud after the second gap detection. (a) Before gap detection and point cloud completeness. (b) After adding the auxiliary images (red) and modelling	145
7.23	Dense point cloud illustrates the gap in different parts of the church before (left) and after (right) gap detection and point cloud completeness.....	146
7.24	The complete point cloud of the church from a layman camera height	152
8.1	2D plan illustration to capture images with the suggested guidance	138

List of tables

4.1	Summary of results	72
5.1	The processing time for the full pairwise and the guided matching	93
5.2	Comparison between different guided matching operators	94
6.1	Pseudo code for the visibility analysis of empty voxels	114
7.1	RMSE of the GCPs and checkpoints	136

Chapter 1 Introduction

1.1 Preface

The creation of realistic 3D models and their visualization is becoming more and more popular. They are a part of the modern digital age: see virtual realities, cultural heritage documentation, geospatial applications, 3D games, TV/movie post-production and special effects (Quan 2010).

Image - Based Modelling is the technique of creating 3D models by the utilization of 2D images (Remondino and El-Hakim 2006). This terminology is well known in computer vision and close range photogrammetry.

What is of interest in photogrammetry is to reach high precision when compared to machine vision (Forstner 2002) while machine vision techniques focus on putting less limitations (for instance the use of uncalibrated camera) and less cost and to create those 3D models in automatic ways.

However, traditional photogrammetric techniques besides using pre-calibrated cameras (known interior orientation) are relying on good approximations of the unknown exterior orientation parameters. This is the reason why tie points can be found using relatively simple (feature-based or area-based) matching techniques. This is because of the significant reduction of the search space for corresponding features. The same holds for the nonlinear least-square bundle adjustment where the equation system needs the approximations of the unknowns.

In computer vision, the calibration of interior parameters is seen as an integrated part of the entire workflow of orientation. This is accomplished by using structure from motion (SfM) technique (Dellaert et al. 2000; Hartley and Zisserman 2004). SfM results in the self-calibration information, cameras exterior orientation parameters, and the 3D point coordinates of the tie points in the object space. However, this fully automated SfM technique doesn't account for the true scale and datum without additional input like ground control points (GCP).

The automated orientation of images in SfM is basically dependent on digitally detecting the homologous points in the overlapping images. Moreover, SfM relies on the computation of the fundamental matrix (F) and RANSAC (Hartley and Zisserman 2004) which is comparable to the stereo relative orientation with coplanarity in photogrammetry (McGlone et al. 2004).

The imaging can be either static (still imaging) or moving (video sequence). With video short baseline imaging the correspondences between images can be found by the so-called *feature tracking* like by using KLT method (Tomasi and Kanade 1992). The advantages of video imaging beside the flexibility of the recording is that the search for frame-to-frame correspondences are based on the assumption that between two consecutive frames corresponding features are hardly moving. Together with some more requirements like the mentioned F-matrix estimation, adding more frames to the solution using resection, some advanced self-calibration technique and finally a bundle adjustment, we are able to reconstruct a scene from a video sequence up to an unknown scale (Nister 2001; Pollefeys et al. 2004). A spatial similarity transformation is normally used to reconstruct the models in the object coordinate system.

The insufficiency of using video image sequence when creating a 3D model is to have short baselines between images. This short baseline can lead to a small Base/Depth (B/D) ratio. The small B/D ratio is geometrically undesirable because of the bad intersection angle which results in a large depth error as shown in Fig.1.1.

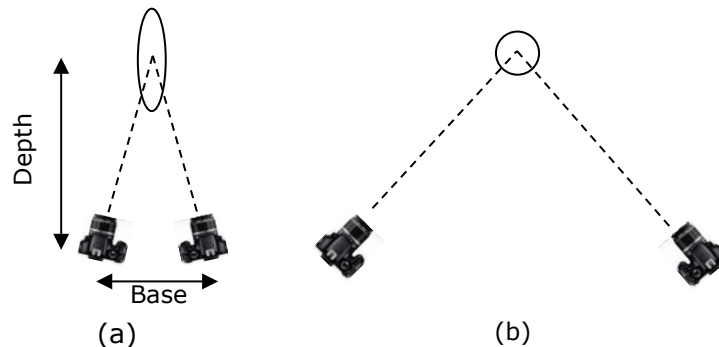


Fig.1.1: Imaging geometry. (a) Small B/D ratio and large error. (b) Large B/D ratio and small error.

In addition the resolution of the current consumer video cameras is not sufficient when compared to the high resolution (HR) still frame images. Moreover, a significant number of the video frames are relatively blurry due to the motion of the camera. Therefore, the created 3D models from video frames are of low geometric and radiometric quality and the better choice is to use a camera with HR still imaging.

For illustration, two 3D models of the same building are produced from video imaging and still imaging in Fig.1.2. The model produced from video imaging looks deteriorated and the facades are not perfectly planar compared to the model produce from still images.



Fig.1.2: Automated 3D model from (37) still images (left) and 3D model from (178) video frames (right).

However, capturing thousands of high resolution still short baseline images to have a complete 3D model is not practical and expensive in the sense of computations. Also, there is no guarantee to achieve the required accuracy and full coverage. Therefore wide baseline images are to be captured. Consequently, corresponding features are probably not nearby in adjacent images, and the image overlap and connectivity is unknown as well. In computer vision approach and in contrary to the aforementioned approach of photogrammetry, there is a lack of the pre-knowledge about the scene and the EO. This would result in too many false matches if area based matching is used because of all the ambiguities and the computational complexity since the search space cannot be reduced. Much better image geometry can be exploited if it would be possible to find the matches in those wide baseline images independent of any pre information.

The scale invariant operators like SIFT or SURF (Lowe 2004; Bay et al. 2008) provide the state-of-the-art methodology for a solution to the wide baseline image correspondence problem. The key solution with these operators is in the efficient histogram based description of (128 or 64 vector elements) of the detected keypoints in the sense of invariance to scale, rotation and illumination.

The image correspondence is computed from a brute-force matching (all images against all), and the so-called matching connectivity tree is computed. However, these techniques are not only expensive in terms of computation time, but also vulnerable to problems of mismatches.

The failure of SfM can be analysed as a result of several causes like the effect of repetitive patterns or symmetries in the buildings architecture (Kosecka and Zhang 2010; Cohen et al. 2012; Wilson and Snavely 2013). These repetitive patterns can mislead the connectivity between images in the full pairwise image matching as shown in Fig.1.3.

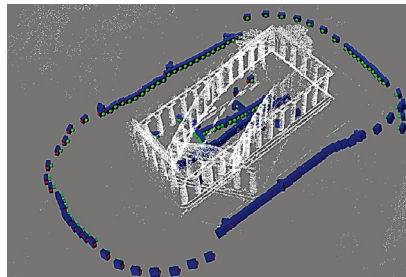
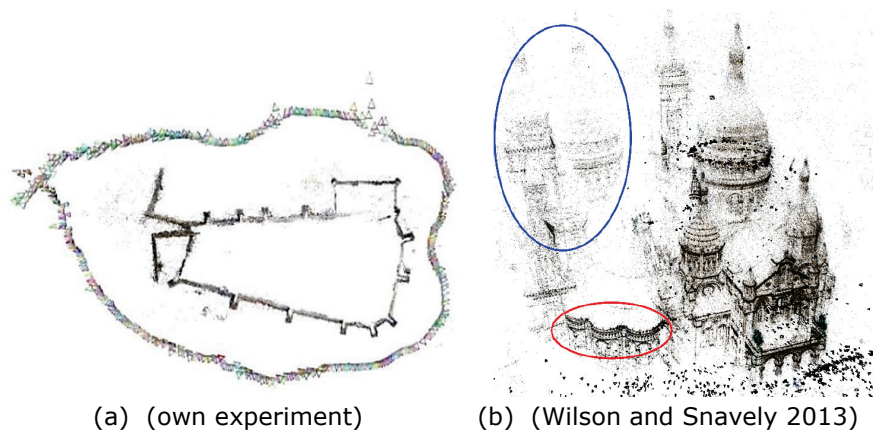


Fig.1.3: Repetitive texture patterns affecting the tie points matching and leading to wrong correspondences.

Furthermore, the SfM failure can result because of a weak or bad planning for the camera network around the object. This improper planning usually results in wide baseline imaging configuration (large B/D ratio). Although, wide baseline imaging gives higher positional accuracy as mentioned previously, the matching of markerless tie points during the SfM computations cannot handle images with wide baselines.

The matching of wide baseline images is still a challenging problem due to the existence of occlusions, affinity, scale change and difference in perspective between the images. Therefore, fully automated orientation of this kind of wide baseline imaging is still a difficult task and manual interference might be necessary (Bartelsen et al. 2012).

This mismatching will probably affect the SfM results. Fig. 1.4a and 1.4c shows the misclosure in the camera network and the effect on the resulting point cloud. Fig. 1.4b shows the repeated and shifted sparse point cloud inside the circles which is resulted because of the mismatching caused by the repetitive patterns.



© (Remondino et al. 2012)

Fig. 1.4: Failure examples of SfM.

To get a more accurate and reliable orientation and then a sufficient 3D model, it is important to place the camera in the proper location and attitude that satisfies all the geometric demands for the coverage and the accuracy of points. This is a preliminary step, but still highly affects the results due to the deep knowledge and cost needed for a good design (Remondino and El-Hakim 2006). The camera planning will also assist to avoid the problem of repetitive pattern in image correspondence and provide a good approximation of camera orientation for running the bundle adjustment.

After the successful step of image orientation with SfM, the technique of dense matching is often applied. The dense matching algorithm attempts to establish a maximum number of pixel-to-pixel correspondences in two images (Hirschmuller 2008; Furukawa and Ponce 2009). The forward intersection of all these points will lead to a dense point cloud in the object space.

Dense matching or 3D reconstruction needs a sufficient percentage of overlap and similarity between the images and good B/D ratio for high depth

accuracy (Remondino et al. 2013) as shown earlier in Fig.1.1. Hence, the imaging configuration is to be designed carefully to find a compromise between the geometry, recovery and correlation (Hullo et al. 2009; CIPA 2011; Haala 2011). Fig. 1.5 illustrates that even with a successful orientation of wide baseline images, the completeness of the 3D point cloud is not achievable. Therefore a compromise between the short baseline and wide baseline imaging is needed.

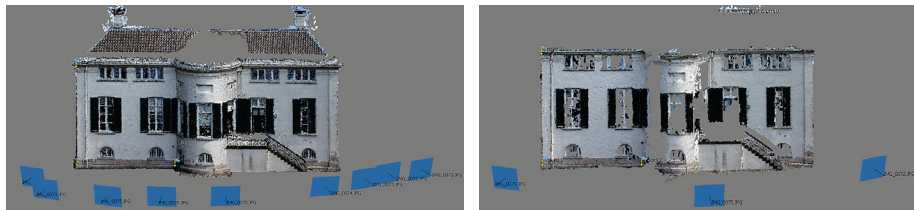


Fig.1.5: Façade point cloud from the short baseline (left) and façade point cloud from the wide baseline (right) using Agisoft.

The image based modelling workflow also contains the modelling of these dense point clouds into a surface mesh (Bernardini et al. 1999; Kazhdan et al. 2006). This is followed with *Texturing* for the reality and aesthetic visualization, which is important in many applications like cultural heritage documentation.

1.2 Research problem

Image-based modelling as stated previously is an important tool nowadays for realistic 3D modelling and other applications in close range photogrammetry and computer vision. Currently, different State - of - the - art software are available for performing the image orientation task automatically by using the SfM techniques. Nowadays, this automatic orientation task can be done with commercial software like PhotoModeler scanner (PhotoModeler 2009), Pix4D (Pix4D 2013) and Agisoft (Photoscan 2011) or open-source like Bundler (Snavely 2010), APERO (Pierrot-Deseilligny 2012) , VSfM (Wu 2012), and 3DF Samantha (3DFLOW 2012).

The orientation will allow creating the sufficient dense 3D point cloud and/or model of the study object by dense matching (Hirschmüller 2008; Furukawa and Ponce 2009). For open source software a dense matching software is needed to process the output of the orientation data to create the 3D dense clouds like PMVS (Furukawa and Ponce 2010), SURE (Wenzel 2013) and MICMAC (Pierrot-Deseilligny 2012). Promising examples are introduced to the community based on internet data sets which consumes a huge power of computing and automation like the work of 'Building Rome in a Day' (Agarwal et al. 2011). The 3D point cloud from a visualization viewpoint looks

interesting although it might not be sufficient for other applications like tangible cultural heritage documentation, large-scale mapping and close range photogrammetry. This insufficiency is due to the lack of completeness of the final 3D model and the probably low predicted positional accuracy compared to the standards of the detailed documentation (Letellier 2007).

The reason is that this huge data set of collected images might be taken randomly without any pre planning. Therefore, reaching the required level of completeness and quantification will be a matter of coincidence.

The key idea of these automated approaches is to have a successful image orientation by SfM and then an efficient dense image matching. However, there is no guarantee concerning the correct results of orientation and modelling completeness and often the reliability is very low (Remondino et al. 2012) especially in the case of large data sets, wide baseline configurations (Fig. 1.5), illumination changes, or when repetitive pattern and homogenous texture areas exist (Fig. 1.4).

Accordingly, the demand of detecting image points and matching them with other correspondences in other images represents a challenging task or the bottleneck in the whole procedure of orientation (Yang and Yuille 1995; Remondino et al. 2012).

Despite the developed SIFT GPU implementations (Wu 2012), matching is still a time consuming operation when a large dataset of captured images n are:

- of high resolution, which is the case nowadays even with consumer compact cameras or smart phones.
- arranged in sparse block where the computation cost according to Barazzetti et al. (2010) is $O(n^2)$ with a combination of images $\binom{n^2-n}{2}$.

From the previous discussion, several shortcomings and problems in the current workflow of the image based modelling are still open and need to be solved. These shortcomings are mainly related to the possible unsuccessful image orientation.

From the previous discussion, the problems to create a reliable 3D image based model can be:

- The short baseline high resolution images can give a better coverage of the model, but to capture and process them is expensive. Moreover, they are unreliable in terms of accuracy as previously mentioned in context with the video imaging.
- The wide baseline HR images are better in terms of positional accuracy and processing time. However, there is a difficulty to have a successful image matching and full coverage of the model.

- The difficulty even for professionals to capture the proper images of an object for 3D modelling. This is because of several reasons like the overlap between the images, the camera viewing angle, the distance to the object and the camera locations.
- The unsuccessful tie point matching which results in a wrong image orientation. Repetitive patterns as mentioned before can lead to wrong matches, especially in wide baseline imaging configurations.
- The difficulty to have a complete 3D point cloud of an object. This is because of the complexity of the object, the texture less parts, and the improper camera planning.

The application area of this research is about the tangible cultural heritage documentation of buildings and sculptures. Nowadays, different countries are investing in the 3D documentation or cataloguing of their cultural objects. However, the 3D modelling of cultural heritage is still not the conventional and a common procedure for such documentation. Reasons are high cost of 3D, the difficulties in achieving good 3D models by everyone, and the consideration that it is an optional method of understanding (an additional "aesthetic" aspect) and documentation (Remondino and Rizzi 2010).

For archaeologists, it is important to find a practical, efficient, low cost, and easy way to construct accurate and well-textured 3D models of tangible cultural heritage objects (finds, free forms, sculptures, carved walls and buildings). In virtual museums, the 3D knowledge is useful for academic and educational projects and culture resource managing (Gilboa et al. 2013). Two valuable examples are the virtual museum of Iraq (Mattei 2008) and the virtual Hampson Museum (Virtual Hampson 2010). The documentation with 3D image based modelling is also valuable when archaeological or cultural heritage sites are preserved in a good way but hardly accessible by people (Stanco et al. 2011).

Fig.1.6 illustrates the pipeline of imaging reconstruction techniques of the reality-based 3D modelling of cultural heritage.

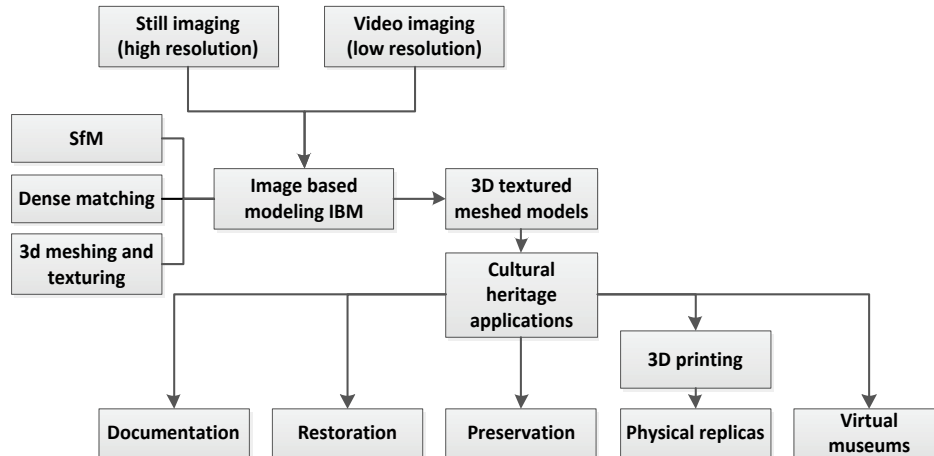


Fig.1.6: 3D modelling of cultural heritage by imaging techniques.

1.3 Research objectives

For the purpose of having highly detailed 3D models and to avoid the shortcomings mentioned in the previous section, the main research objectives will be as follows

- 1- Optimal camera network design: The first objective is to find the suitable camera placement where high resolution images will be captured. The imaging network should be designed with a balanced number of images and compromise between the short baseline and wide baseline imaging configuration. Moreover, geometrical constraints like reliability and visibility need to be satisfied.
- 2- Guided imaging system: This objective is to enable the camera operator to find the suitable placements around or inside the study object and capture the images. The major task is the design of a user-friendly guiding tool. This will enable even non-professional persons to interactively understand and realize the capturing of the images at/on the site.
- 3- Image orientation with guided tie points matching: This objective comprises the guidance of tie points matching between the captured images. This guidance should include the connectivity between the images and between the keypoints as well.
- 4- Gap detection to check for the completeness of the 3D model: This objective aims to find the possible supplementary images that can complete the coverage of the missing parts of the object point cloud.

The following questions must be answered to achieve the research objectives:

1. How we can design the camera network?

In order to find the camera placements that guarantee the optimal coverage and accuracy, the following questions should be answered

- I. How to retrieve relevant geometric information of the target object?
- II. What are the necessary constraints that lead to the optimal network geometry and camera placement?
- III. What is a suitable optimization technique for the camera network design?

2. How to build the guiding system?

- I. How to indicate the proper guided location?
- II. How can the system guide the camera operator to the correct location?

3. How to guide and improve the tie points matching?

- I. What is the most suitable and efficient tie point matching and orientation technique?
- II. How to exploit the knowledge on image overlaps to avoid using the full pairwise matching?
- III. How to verify the accuracy of the actually captured imaging network against the accuracy of the designed one?

4. How to ensure the completeness of the 3D model?

- I. How to detect gaps in the point cloud automatically?
- II. Where to capture the auxiliary images for the final modelling?

All these questions will be answered through the proposed research. The general summary of the research objectives and questions is illustrated in Fig.1.7.

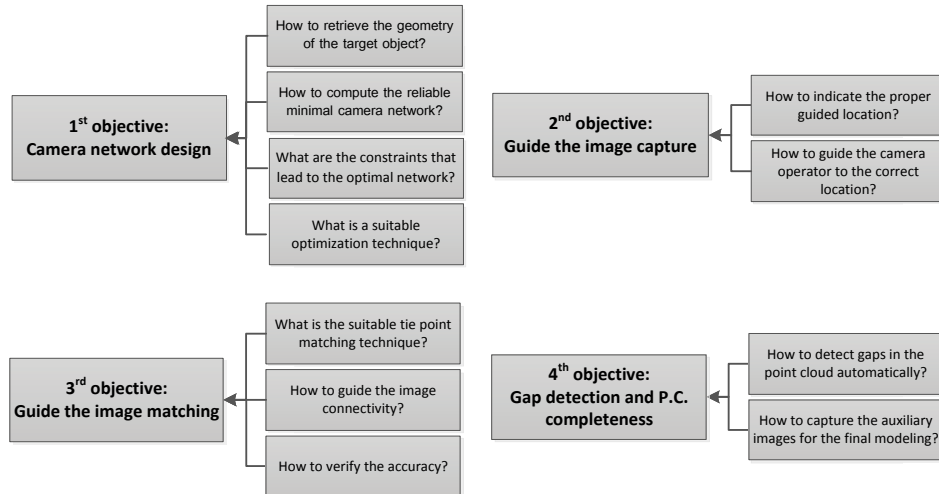


Fig.1.7: The summary of research objectives and questions.

1.4 Scope and limitations

This research aims to build an imaging system that guides the camera operator, even a non-professional, to capture the images in a simple user-friendly way. The research scope is developed mainly for the documentation of tangible cultural heritage objects. More specifically, the workflow has been developed for image based modelling of buildings and statues, but is not limited to those applications. The research assumptions are based on conducting the imaging with a non-metric hand held camera. The camera can be pre calibrated or uncalibrated where a self-calibration is needed during the image orientation.

Furthermore, the research methodology is based on applying terrestrial close range imaging that is restricted to the accessible heights by a layman.

1.5 Innovations in this work

This research investigates the ability to use automated image based techniques in creating a complete and reliable 3D model. Different methods will be developed to accomplish this modelling aim. A simulation of a dense imaging network followed by a reduction of imaging locations leads to the minimal camera network. Then mathematical optimization techniques are used to find the optimal imaging configuration that offers higher accuracy than the pre-designed network. Robust correspondence matching is also to be developed. Finally, we want to automatically detect the possible gaps in the 3D dense point clouds and to recover them.

The innovation of this work is: firstly, automating the camera network design that ensures both, coverage and accuracy requirements for the 3D modelling. Secondly, guiding the cameraman to the designed locations by a user-friendly and easy to implement approach.

Thirdly, a robust image correspondence matching is to be found relying on the camera network design and the object rough model. Finally, the possible gaps in the constructed 3D models are to be detected automatically and then to be recovered by capturing new auxiliary images. This is to be accomplished either by using image processing techniques or by voxelization approach.

1.6 Thesis Outline

The second chapter presents a literature review on the state-of-the-art in the Image based modelling and camera network design and optimization. The problem of visibility analysis of point cloud is also addressed. The third chapter summarizes the general methodology. The fourth chapter describes the camera network design, the minimal network and the optimization of the networks.

Chapter five describes the guiding system design for capturing the images in the field as well as the computation of the image connectivity graph. The correspondence matching is also investigated for the final accurate image orientation.

Chapter six presents the occlusion problem in 3D modelling and the procedures for detecting the resulted gaps in the point clouds. The final image capture and 3D modelling is also described.

Chapter six will present experimental tests that show the impact of using the proposed methodology in the application of cultural heritage documentation. The last chapter describes the conclusions and recommendations for future research.

Some parts of this thesis are quoted from papers published during the research (Alsadik et al. 2012; Alsadik et al. 2013a; Alsadik et al. 2013b; Alsadik et al. 2014a, Alsadik et al. 2014b).

Chapter 2 Literature Review

The overall image-based 3D modelling (IBM) process consists of several successive steps (Remondino and El-Hakim 2006) as follows:

- Design (sensor and network geometry).
- 3D measurements (sparse and dense point clouds).
- Structuring and modelling (segmentation, network/mesh generation).
- Texturing and visualization.

The previous research work that is dealing with the mentioned topics of IBM, will be discussed beside the state-of-the-art in this domain.

2.1 Design (sensor and network geometry)

The main purpose of network optimization is to design an optimal network configuration (how many stations are needed) and optimum-observation plan (how many images per station are needed) in the sense that they will satisfy the network quality requirement at a minimum cost (Kiamehr R. 2003).

During the last three decades, the topic of network optimization was one of the interested and challenging problems in geodesy, photogrammetry, machine vision, and robotics.

Olague (2000) defined the photogrammetric network design as the process of placing a camera in order to perform photogrammetric task. From a geodetic viewpoint, Graferend and Sanso (1985) classified the design into four different problems:

- Zero Order Design (ZOD) – the datum problem
- First Order Design (FOD) – the configuration problem
- Second Order Design (SOD) – the weight problem
- Third Order Design (TOD) – the densification problem

In the ZOD problem, the aim is to define an optimal datum to obtain accurate object point coordinates and exterior orientation parameters. Mathematically the design matrix has a datum rank defect and some form of free adjustment approach through generalized inverse is adopted (Fraser 1984). The aim of the FOD stage is to define an optimal imaging geometry which means the determination of the camera orientations that lead to the desired reconstruction accuracy. The aim of the SOD is focused on adopting a suitable measurement precision for the image coordinates (weights) while TOD deals with the improvement of an existed network through the addition of new observations to the weak parts of the network (Fraser 1982; Fraser 1989).

Although the design problems were mainly for geodetic networks, photogrammetrists were also in need to optimize the imaging networks to get

better accurate positioning results. However, it was more difficult to handle these 3D networks.

Fraser (1982) pioneered the network design application in close range photogrammetry by trying to study the effect of minimal constraints and free adjustment of the networks. Later on, Fraser (1984) investigated the design of non-topographic photogrammetric networks. The introduced study was based on a network simulation and results on specific design characteristics that is widely used nowadays.

Mason (1995, 1997) presented a novel technique for designing the imaging networks in close range photogrammetry. He built a system called CONSENS (CONfiguration of SENsor Stations). The system was based on what is called generic network, which consists of four intersected images to provide a strong imaging geometry. Mason also defined the workspace of the camera in a tessellated sphere to limit the searching space. The experiments were always on a planar surface rather than complex shaped objects.

From the CV community, a system called EPOCA (an acronym for "Evolving POsitions of CAMeras") was introduced in (Olague and Moher 1998), (Olague 2002),(Dunn et al. 2006) and (Olague and Dunn 2007). They divided the problem into two main components: analytical part dedicated to the analysis of error propagation from which a criterion is derived, and a global optimization (genetic algorithm) process to minimize this criterion. Their work concentrated on how to solve the problem of optimum camera placement mounted on robots in a limited workspace of a lab.

An automatic sensor placement method in the field of vision metrology based on fuzzy logic method (artificial intelligence) was introduced by (Saadatseresht et al. 2004) ,(Saadatseresht et al. 2005), (Saadatseresht and Varshosaz 2007) and (Amini A. Sh. 2010). The concept was based on the challenge of designing the imaging network without pre knowledge about the object workspace. The method depends on taking many images from different views of the study object and was then followed by automatic enhancement of the accuracy of the existing network by adding new exposure stations.

In active vision the camera placement is determined in automatic way (robot work cell) for the purpose of having valid viewpoints towards the moving objects like in surveillance planning (Abrams et al. 1996). Yang and Ciarallo (2001) tried to find the optimized camera placement for dimensioning a set of edge segments in a pre-designed CAD model of simple industrial objects by using genetic algorithm technique (Yang and Ciarallo 2001).

2.2 Three Dimensional Measurements And Reconstruction

The use of images in the reconstruction of 3D models, requires different interrelated techniques as will be discussed in the following subsections.

2.2.1 Structure from Motion (SfM)

Structure from motion can be defined as the automated process of retrieving camera orientations beside the 3D positions of the tie points by analysing an image sequence.

This is an important theme in computer vision, and great progress has been made both in theory and in practice during the last two decades (Hartley and Zisserman 2004; Quan 2010; Wang and Wu 2011).

The SfM pipeline consists of a process to find correspondences along the stereo images using point features extracted by operators like SIFT (Lowe 1999; Lowe 2004) or SURF (Bay et al. 2008).

A following step is to estimate the fundamental matrix of each stereo pair with the use of outlier detection of RANSAC (RANDOM SAMPLE CONSENSUS) as mentioned in chapter 1. The interior orientation parameters of the camera are necessary to compensate the lens distortion and to have a metric reconstruction by computing the so called essential matrix (Hartley and Zisserman 2004). Then, the orientation (projection) matrix is computed for every image and a bundle adjustment for the whole block is to be carried out to ensure optimal results (Lourakis and Argyros 2004; Pollefeys et al. 2004). Bundle adjustment is a well-known mathematical technique in photogrammetry and computer vision. The concept is to minimize the errors between the computed and the observed image coordinates. Least square adjustment is to be used for the minimization either by using Levenberg – Marquardt or the Gauss-Newton method.

Nister (2001) investigated the use of video imaging for a reality-based 3D modelling by using SfM technique, followed by a dense matching (section 2.2.2). The models created by Nister were not so much realistic because of the limitations of Delaunay mesh and the known low resolution of the video camera as mentioned earlier in chapter 1.

Remondino and El-Hakim (2006) reviewed the efficiency of using the camera self-calibration (with or without control points). They advised to use this technique only when the following is available: a strong geometric configuration with convergent images and a large number of spatially well distributed (3D) targets.

Recently, research work (Cohen et al. 2012; Wilson and Snavely 2013) is conducted to avoid the failure case of SfM because of the repetitive patterns or structures that is mentioned in chapter 1.

2.2.2 Dense Matching

The construction of a highly detailed and fairly realistic textured 3D model, is possibly based on creating a dense point cloud which necessitates to implement a dense matching between oriented images.

Conventional techniques depend on correlating small image windows along the one dimensional epipolar line within the maximum disparity range (Lhuillier and Quan 2002).

However, these methods assume constant disparities within a correlation window, which is incorrect at discontinuities and leads to blurred object boundaries (Hirschmuller 2005).

Besides the strategy of limiting the search to the epipolar line, there are also the strategies of working with hierarchies of image pyramids, matching feature and edges first to provide better approximations for successive area-based matching (ABM), using a region-growing from initial seed points, and computing intermediate depth maps (Remondino et al. 2008).

Pollefeys (2008) used the plane sweeping algorithm for computing the depth maps. A fusion for the successive depth maps from image pairs is to be done to reach the final 3D model for 3D urban mapping from a moving vehicle.

These mentioned techniques are not always effective especially in a wide baseline convergent imaging due to perspective distortions and possible occlusion problems (Remondino et al. 2013).

The advanced techniques in dense matching are aiming to have an accurate dense stereo matching, especially at object boundaries, robust against illumination changes, detect occlusions, and determine disparities with sub-pixel accuracy (Hirschmuller 2005).

Two of these advanced techniques which are based on formulating the matching as an optimization problem are the dynamic programming (DP) technique and the Semi Global matching (SGM).

In dynamic programming the effect of discontinuities is reduced by pixelwise matching but requires other constraints for piecewise smoothness. This means to constrain the disparities to be changed only by a certain amount

between adjacent pixels. The scan lines search limited to one dimension (1D) and independently for each pair of scan line which leads to streaking effects (Lysak and Kapshiy 2008).

Hirschmüller (2005,2008) introduced his novel method of Semi-Global Matching (SGM) method. The SGM method uses a pixelwise Mutual Information (MI) cost metric and approximates a global 2D smoothness constraint by a combination of multiple 1D constraints.

An additional constraint is added to the cost that supports smoothness by penalizing changes of neighbouring disparities, one penalty for the little changes (below 1 pixel) and the other for the larger disparity changes. Then the optimization is aggregated through the combination of several 1D optimizations.

Another dense matching strategy depends on using the initially matching feature points as anchors for computing the full reconstruction. The disparities of the matched features points are propagated to their neighbours while uses them to initialize an iterative estimation of the depth maps (Tola et al. 2008).

In a similar way (Furukawa and Ponce 2009) produced the state-of- the-art Patched Multi View Stereopsis (PMVS) algorithm. It consists of three successive steps which are matching, expanding, and filtering. In the first step corners are detected in each image and followed by matching the homologous points along the epipolar lines. The computed 3D coordinates of these matched points represent the initial patches. The expansion step is to search the neighbouring image pixels and assign them as new patches after implementing a decision strategy to avoid overlaps and discontinuities. The normal and centre of each expanded patch can be estimated and optimized. The final step is to remove the wrong patches by applying three filters relying on the visibility consistency. Accordingly, PMVS provide with fine surface details despite low-texture regions, large concavities, and/or thin, high-curvature parts. Optionally, the resulting patch model can be turned into a triangulated mesh.

2.2.3 Modelling and Texturing

Meshing and Texturing are the last steps in the workflow of the 3D reconstruction or IBM. Different techniques can be used to create a reality-based 3D model starting from a dense point cloud which can be classified into three main techniques:

- Volumetric representation (voxels).
- Surface representation (polygonal mesh).

- Point patching techniques.

Volumetric representation (use voxels as unit volume elements) was first introduced in the early 70's in medical imaging and now it has been commonly used in scientific visualization, computer graphics and computer vision. The disadvantage of this technique is the need for a huge amount of memory and processing although it gives a better representation for the complex objects (Kuzu 2004).

Remondino and El-Hakim (2006) stated that polygonal meshing is usually the most flexible way to accurately represent the results of 3D measurements, providing an optimal surface description. One of the most popular polygonal 2D mesh algorithm is the *Delaunay triangulation* method while mesh surfaces in 3D space are called tetrahedralisation. Generally, these methods need a starting point like visual hull model, the computations of additional information like vertex normal, and an adequate number of points.

Two effective and widely used 3D surface meshing techniques are the Poisson method (Kazhdan et al. 2006) and Ball pivoting method (Bernardini et al. 1999).

After constructing the 3D mesh it can be assigned a photorealistic appearance by the process of *Texturing*. Remondino and El-Hakim (2006) illustrates that for texture mapping, the colour images are mapped onto a 3D surface. Knowing the parameters of the interior and exterior orientation of the images, the corresponding image coordinates are calculated for each vertex of a triangle on the 3D surface. Then colour RGB values within the projected triangle are attached to the surface and the final product is a 3D textured model.

2.3 State-of-the-art software in reality IBM

It is useful to review the state-of-the-art software that is currently available for the users in the context of the reality IBM. The currently available software is either commercial, open source or a web application.

Generally, open source software is available for solving a part of the whole IBM workflow like the task of image orientation with SfM.

A SfM solution for a large data set of unordered images collected from the internet was introduced by (Snavely 2008) and the result was the State-of-the-art open source software called *Bundler* (Snavely 2010). Bundler processes a set of images, image features, and image match as input, and produces a 3D reconstruction of the camera poses and a sparse scene

geometry (point cloud) as output relying on a modified version of the Sparse Bundle Adjustment SBA (Lourakis and Argyros 2004). Bundler is also used within the Microsoft web-based application of Photosynth (Photosynth 2012).

Visual SfM software by Wu (2012) offered the implementation of SIFT GPU which extremely speeded up the tie points matching. Moreover, this open source software has many other abilities like the possibility to guide the connectivity between the images and image correspondences as well.

Another open source software designed for the automation of IBM is called MICMAC (Pierrot-Deseilligny 2012). MICMAC is a complete package of tools that can be used to solve the image matching (TOPIACA) and orientation (APER0). Then, to create a dense point cloud. However, the current version still runs on CPU which consumes more processing time.

For dense matching task, open source software like SURE (Wenzel 2013) which is based on using SGM and PMVS (Furukawa and Ponce 2010) are currently quite promising and widely used.

The interesting web service 123D catch (Autodesk 2012) which is mainly released for 3D printing services, offered a complete package of image orientation and 3D modelling as well. However, the resulting accuracy is not suitable for close range photogrammetry applications (Chandler and Fryer 2011). The first web service of 3D IBM was the ARC3D(ARC3D 2006) which is initiated by Vergauwen and Gool (2006). Currently, a web-based service similar to 123D catch is released by Autodesk company which is called ReCap 360 (Autodesk 2013). ReCap 360 is designed to use the full resolution of the uploaded images, however, the free version is limited to process at maximum only fifty images. This IBM is released with another non-free standalone software version of laser scanning data called ReCap Pro.

Among the known State-of-the-art commercial software are Photomodeler (EOSsystems 1994), Agisoft Photoscan (Photoscan 2011) , Pix4d (Pix4D 2013) and Zephyr (3DFLOW 2013).

Photomodeler is the pioneer commercial software for close range photogrammetry that has the ability to use the images for either manual, semi-automatic or fully automatic 3D modelling. A camera pre-calibration ability is also offered with coded target sheets. Another closer range photogrammetry software is the Australian software iWitness (Photometrix 2003).

Additionally, Pix4D software is currently the state-of-the-art for processing the Unmanned Airborne Vehicle (UAV) imagery. Agisoft Photoscan is also

widely used nowadays for automated camera orientation, orthophoto production and 3D modelling. Lately, another software called Smart3DCapture (Acute3D 2013) is released with two versions of *scanner* edition which handles up to 10 gigabytes of data and *mapper* edition which can handle unlimited number of images for large scale mapping projects . On the other hand, different software is available in the market that is capable of creating 3D models either as rendered graphical or as mapping textured models. As an example: Meshlab, 3D Reshaper, Maya, 3Dstudio MAX, LightWave, Polywork, and Rhino.

Fig.2.1 summarize and classify the current state-of-the-art software for the IBM.

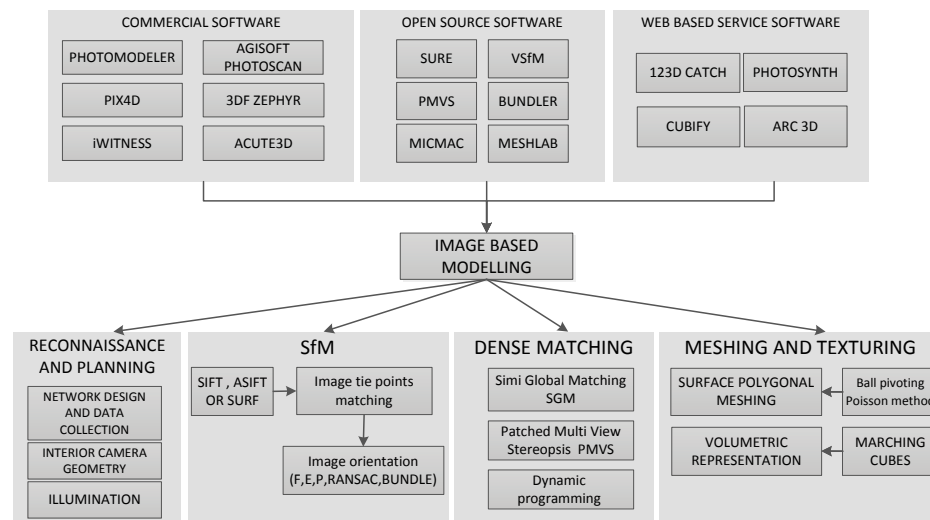


Fig.2.1: General techniques and state-of-the-art software for IBM.

2.4 Shortcoming in the IBM State-of-the-art

As mentioned in the previous section 2.3, a wide range of commercial and open source software is available for users to create their 3D image based models. However, as indicated in chapter 1, there are several shortcomings and challenges in the current workflow that can be summarized as follows:

- Image capture and planning: a significant difficulty is to capture the proper images for the task of 3D modelling. This means a difficulty to determine the necessary number of images to be captured and their orientations that guarantee the full coverage and the required accuracy of the project.
- Tie point matching and image orientation: despite the use of a scale invariant feature operator like SIFT with the GPU implementation (Wu 2012) for speeding computations, the full pairwise matching still comprises difficulties. The difficulties arise with the existence of repetitive

texture patterns like in building facades which likely mislead the SIFT descriptor matching. Moreover, the full pairwise matching is not suitable and expensive for on-the-fly 3D modelling applications like in disaster mapping in the sense of time consumption and computations with a total combination $\binom{n^2-n}{2}$ for n images. For image orientation, the iterative bundle adjustment needs good approximate values for the exterior orientation parameters. Otherwise, the solution might be unstable and diverge.

- 3D model reliability: the reliability of the produced dense point cloud in terms of complete coverage and point accuracy is still questionable. This incompleteness is related beside other factors to the mentioned shortcomings in the image capture plan, the complexity of the study object, and the effect of occlusions.

In this thesis, solutions will be presented for the aforementioned shortcomings as formulated in the research objectives in chapter 1.

2.5 Visibility

Computing the visible part of a 3D point cloud is a vital problem in computer graphics, computer vision, robotics, GIS and photogrammetry. Usually the visibility should be accomplished in an automated way from a certain viewpoint or camera.

The visibility problem is to be handled with both sparse or dense point cloud as shown in Fig.2.2 where the objects looking direction cannot be identified (Katz et al. 2007). This necessitate to use the visibility testing and to discard the occluded points to properly view the object points.

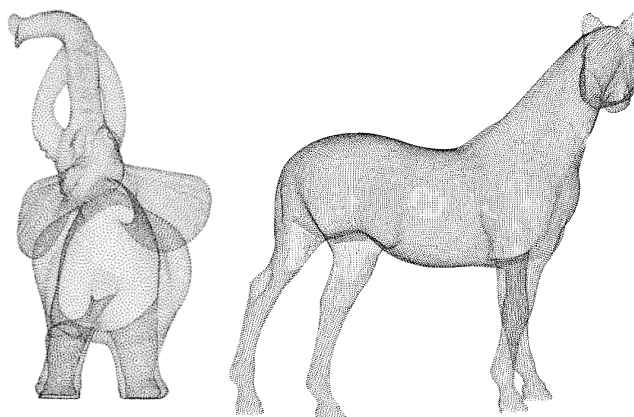


Fig.2.2: Point cloud in an unknown looking position either forward or backward. Sample data quoted from (Luigi 2009).

The earlier digital methods of terrain visibility analysis is presented in GIS as known as viewshed analysis (Yoeli 1985). The method is simply to analyze the Line Of Sight LOS between the observer and the target. This is by comparing the tangent of the LOS angle and the other angles of the terrain points. The visibility is considered blocked when the tangent of the angle between the observer and an middle terrain points is greater than the tangent of the observer-to-target angle (Fisher 1996).

During the last two decades, different methods were developed to solve the visibility problem in the field of computer graphics for real-time rendering and compute games (Bittner and Wonka 2003; Cohen-Or et al. 2003). Currently, the method of hidden point removal HPR (Katz et al. 2007) is widely applied for the visibility analysis. The advantage of this technique is to avoid creating a surface from the point cloud which might be expensive and this led to analyze visibility efficiently with both sparse and dense clouds. However, when the point cloud is noisy or non-uniformly sampled, a robust HPR operator (RHPR) is preferred to be used (Mehra et al. 2010) to deal with these cases.

Other techniques are based on creating a triangulated mesh surface like by using Poisson reconstruction (Kazhdan et al. 2006) or ball pivoting (Bernardini et al. 1999). After we create the surface, the notion of visibility can be uniquely defined and then find its hidden and visible points from any viewpoint. This is mathematically achieved by either intersecting the line of sight rays with the surface triangles or checking the orientation of the surface normal.

With volumetric data applications, a voxel based techniques are suitable more than triangle based. However, we cannot simply adopt those voxel techniques (Kuzu 2004). The surface normal vector is more expensive to compute and less accurate as well. Therefore line tracing and z-buffering is usually used with these volumetric data types.

In line tracing, the concept is to define the ray between the voxel in question and the viewpoint. Then tracing this ray towards the destination and stop when another voxel is intersected.

The z buffering or depth buffering is perhaps the simplest, and is the most commonly used according to Joy (1999).

These visibility methods can be summarized as:

- The surface triangle based methods (the normal direction testing, triangle – ray intersection, Z –buffering method).
- The voxel based techniques (voxel - ray intersection, ray tracing and Z-buffering method).

- The hidden point removal (HPR).

2.5.1 Surface triangulation based methods

The triangulation based methods can be applied by either testing the surface normal direction, intersection between a triangle and a line in space or by using the distance buffering by projecting the points back into a plane (image). These three methods require the creation of a triangulated surface which can be expensive in terms of computations and time consuming.

- A. **Testing the surface normal direction:** This method is considered a simple method when compared to the other two methods since it is just based on testing the angle difference between the vertex (or face) surface normal and the viewing direction. The methodology is based on creating a triangulation surface and computes the normal vector for each vertex or face. Several efficient methods are found for the surface triangulation like ball pivoting (Bernardini et al. 1999) and Poisson reconstruction (Kazhdan et al. 2006). These normal vectors are used to test the visibility of points in each camera as shown in Fig.2.3 which shows a simulated building facade example.

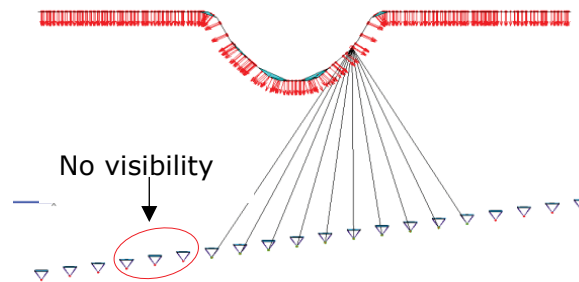


Fig.2.3: Visibility by using the triangular surface normal vectors.

Accordingly, the decision of considering points as visible or invisible is depending on the absolute difference between the orientation of the camera optical axis Cam_{dir} and the face normal direction N_{dir} . This difference is compared to a threshold (like $<90^\circ$) to decide the visibility status.

However, it must be noted that by only using this technique, we are not able to detect and avoid occluded points. This is obvious when the angle difference is less than the threshold while the protrusion of a façade occluding the point.

- B. **Ray - triangle intersection:** This method and the method of ray -voxel intersection is based on the same geometrical strategy where each triangle vertex is tested whether representing the first intersection point

with the line emerging from a certain viewpoint or not. Being not the first intersection point indicates the occlusion case. Every vertex point in every camera or viewpoint should be tested to reach the final visibility labeling. This illustrates the large amount of computations needed in these geometrical intersection methods. However, it seems accurate and no incorrect visibility cases can arise. (Fig.2.4)

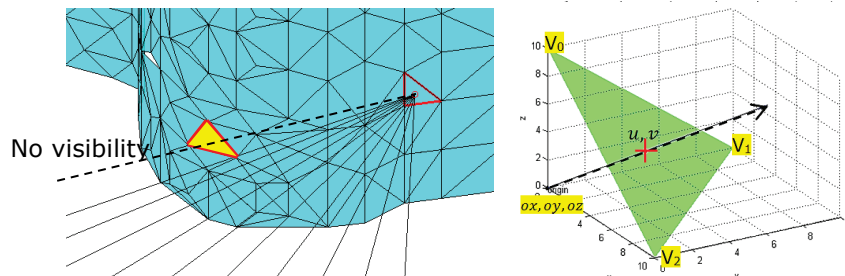


Fig.2.4: Visibility by testing the ray-triangle intersection.

Mathematically, the intersection method is based on calculating the intersection of a line and triangle in space (Möller and Trumbore 1997). However, the disadvantage of this visibility method is the difficulty of the reconstruction (surface triangulation) since it often requires additional information, such as normals and sufficiently dense input. Moreover, it is computationally expensive with a large data set because every triangle should be tested for the intersection.

- C. **Z-buffering method:** The third triangle – based technique is the Z-buffering or depth buffering method which is applied by projecting the surface triangles back to a grid plane like a digital image. These back projected 2D triangles are tested whether represent the closest or the farthest from that plane. The occluded triangles will be neglected and only keep the close triangles which should be visible from the defined viewpoint as shown in Fig. 2.5. The final visibility map is like a digital image, but the pixel values are the (x, y, z) coordinates instead of the RGB information. The pixel size should be selected carefully to avoid extra processing time or less efficient results.

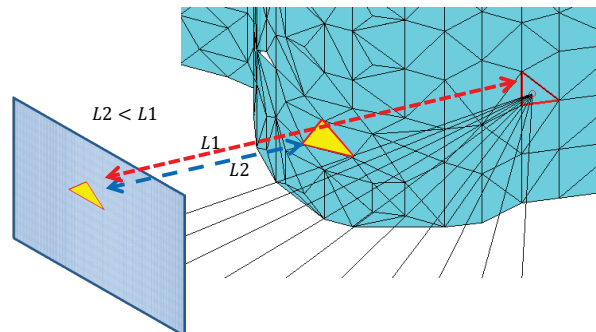


Fig.2.5: Depth - buffering method.

2.5.2 Voxel based approach

In some applications like gaming or other computer graphics applications, the point cloud is represented as voxels and it seems very useful to analyze the visibility on the basis of voxels rather than points. The advantages of using these methods are the avoidance of creating a surface while it is considered an expensive approach in terms of computer memory. Three different techniques are listed in the following sections which are: voxel -ray intersection, voxel distance buffering and ray tracing methods.

- A. **Voxel – ray intersection:** In this technique, the visibility test is applied by intersecting a ray emerging from the viewpoint (origin-o) with a certain direction $d(dx, dy, dz)$ to the voxels (v_{min}, v_{max}) and to check if it intersects (flag=1) or not (flag=0). This is a typical line-box intersection problem presented by Williams et al. (2005) and coded by Mena-Chalco (2010). Turning the point cloud into voxels is simply driven by gridding the space occupied by the points according to a specific voxel size. This is followed by discarding the empty voxels and keeping all the occupied voxels as shown in Fig.2.6.

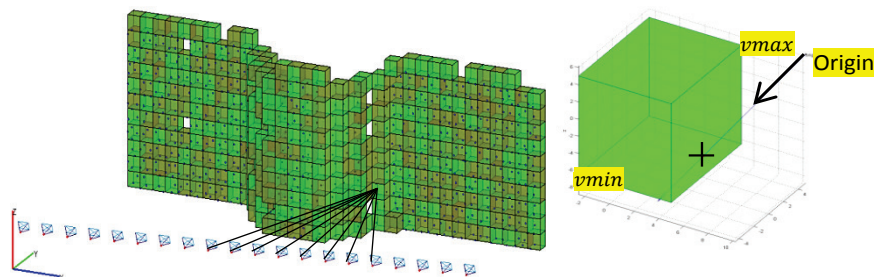


Fig.2.6: Voxel-ray intersection for visibility.

To speed up the computations of the intersection algorithm, a bounding volume hierarchy BV is created (Smits 1998). Mathematically, the intersection involves computing the distance from the origin to the intersection point which object was hit Mena-Chalco (2010).

The advantages of this method beside the avoidance of surface reconstruction are that no settings are required to implement the method except the voxel size to get better accurate results. However, disadvantages of this method arise when processing a large data set because it can be expensive in terms of time and memory consumption.

- B. **Buffering technique:** The distance buffering method is applied in a same way as in the triangle buffering method. Projecting the voxels back to a grid plane and testing whether the 2D polygons represent the closest or the farthest from that plane. The occluded polygons will be neglected while the non-occluded polygons which should be visible from the defined viewpoint will be kept. The time consumption is a main disadvantage of this method.
- C. **Ray tracing technique:** The ray tracing method or voxel traversing is simply implemented by computing tracing points (or voxels) along the ray toward the destination voxel. These tracing points will be computed every small interval which is less than the voxel size. Then tested whether they intersect or hit a voxel before reaching the destination. The voxels will be labeled as visible or hidden based on this methodology.

It is worth to mention that the difference between the ray tracing method and the methods of buffering and ray-voxel intersection is mathematically:

- The ray tracing method is a forward computation starting from the viewpoint position and proceed in specific intervals.
- The ray-voxel intersection is an inverse computation between the voxels in question and the viewpoint.

2.5.3 Hidden point removal (HPR)

The concept of this method is applied by assuming the viewpoint C is placed at a sphere origin. The point cloud is projected through the sphere to the opposite outer side in what is named spherical flipping (Fig.2.7). Spherical flipping reflects a point with respect to the sphere by applying an equation defined by (Katz et al. 2007). The flipped point cloud and the viewpoint will be represented by a convex hull (yellow shaded in Fig. 2.7). Then, the transformed points that are located on the convex hull are extracted as visible points.

The major advantages of this method are to determine the visibility without reconstructing a surface compared to the previous surfacing methods beside the simplicity and short implementation time. Moreover, it calculates visibility for dense as well as sparse point clouds, for which reconstruction or other methods, might be failing. However, the disadvantage is realized when a

noisy point cloud exists (Mehra et al. 2010). Moreover, it is necessary to set a suitable radius parameter that defines the reflecting sphere (Alsadik et al. 2014b).

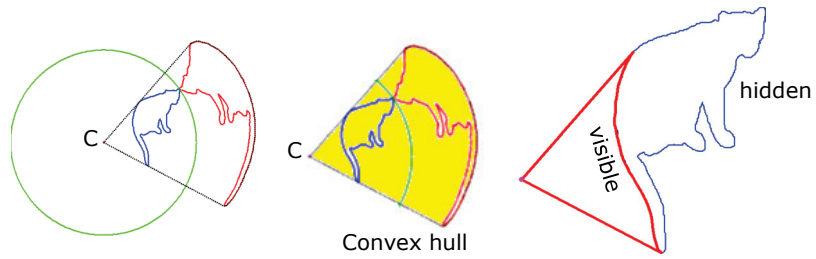


Fig.2.7: HPR method (Katz et al. 2007).

Chapter 3 Overview of Methodology

To give the reader a complete picture about the research implementation in the following chapters, the general methodology of this thesis will be described.

As stated previously in chapter 1 in section 1.2, four research objectives are to be investigated as follows:

- 1- Camera network design.
- 2- Guiding the camera operator.
- 3- Guiding the tie points matching in the SfM.
- 4- Detecting and recovering for the occluded gaps in the 3D point cloud.

These research objectives are to be tested in the application of cultural heritage documentation specifically with statues and buildings. The following sections introduce the general methodology (Fig. 3.1) of this thesis and the details will be investigated in the following chapters.

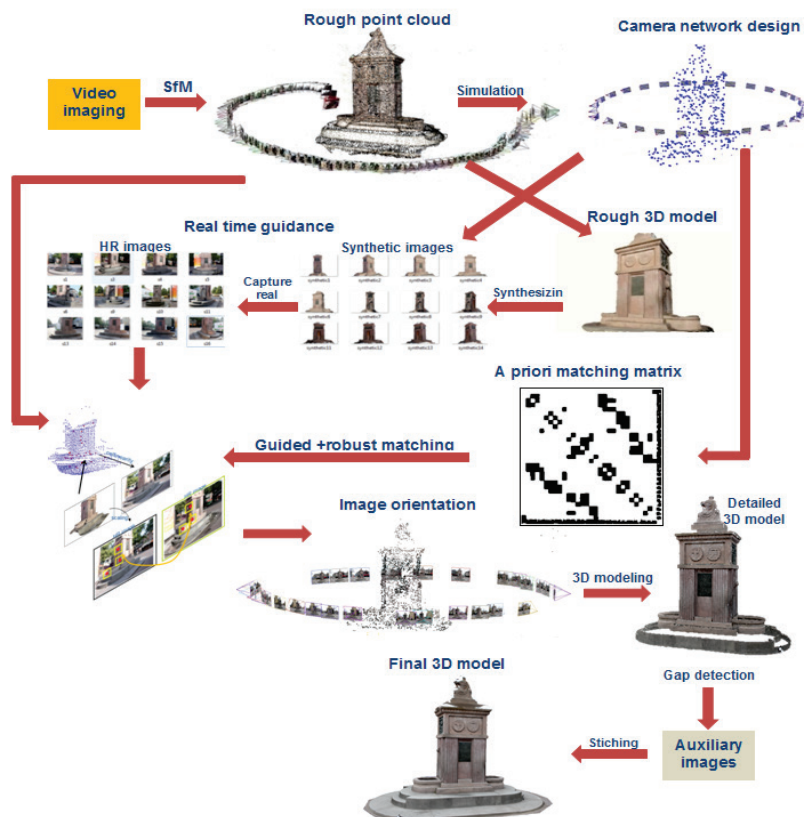


Fig.3.1: The general methodology of the proposed camera network planning and guidance.

3.1 Camera network design

The first objective of this research as will be described in chapter 4 is to build a camera network design system that guarantees sufficient coverage of the object and reliable image-based 3D model reconstruction. Furthermore, the design is expected to reduce the cost and enable non-professionals to create these 3D models.

The initial step to implement the design is to create a rough 3D model of the object to recognize its rough shape and size. This is a necessary input for the subsequent design of a camera network in the sense of coverage and accuracy.

This initial step of rough modelling can be acquired by video imaging of the object where no experience or conditions are necessary during the shooting. Firstly, a dense camera network will be designed according to the geometrical specifications of Ground Sample Distance (GSD), camera interior geometry, B/D ratio and the estimated positional accuracy $\sigma_{x,y,z}$.

Secondly, the dense network will be reduced to the minimal network that still guarantees a full coverage of the whole object. Different strategies of filtering will be proposed and tested. An optional step is to apply optimization techniques to the minimal networks to improve the positional accuracy of the object points with respect to constraints. The general methodology of this objective is shown in Fig.3.2.

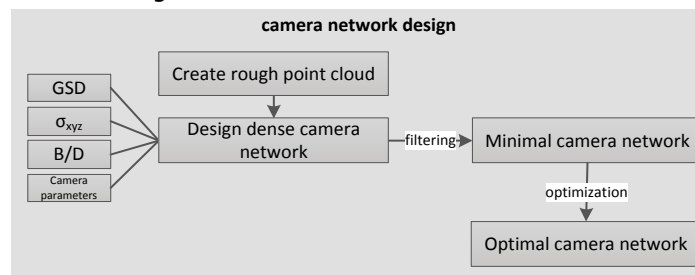


Fig.3.2: The methodology of camera network design.

3.2 Guiding the image capture

General rules were introduced to simplify the capture of the images for the 3D modelling of buildings, statues or complex sites (Hullo et al. 2009; CIPA 2011; Remondino et al. 2013). However, these guiding rules are still difficult to be implemented by non-professionals and sometimes even to professionals. This difficulty is because of the dependency on the object geometries, size, accessible space and to maintain the camera orientations in a specific pose.

Therefore, and in connection with the objective of the camera network design, a guiding procedure is introduced in this thesis in chapter 5.

Accordingly, the designed optimal network is to be captured on the site where the camera operator has to recognize where to locate and how to rotate the camera. The proposed methodology is to synthesize the images that should actually be taken by retrieving the rough 3D model that is produced from video imaging. Then, a SIFT matching between the synthetic and real captured image is to be made. This is necessary to compute the equivalence between the planned and actually taken images. Error estimation of the object points will assist in making the decision if the captured images are sufficient or to recapture the image. Fig.3.3 shows the general workflow to implement this objective.

3.3 Guiding the image tie points matching

To have a successful tie point matching as described earlier in chapter 1 and chapter 2, different conditions are to be met. These conditions might be: the good B/D ratio, good illumination, strong internal camera geometry and the low complexity of the object. Although most of these conditions can be fulfilled, the complexity of the object like the effect of repetitive patterns is still difficult to avoid.

Therefore, and in connection with the previously described objectives, a guiding methodology of the tie point matching will be introduced in this thesis in chapter 5. The method is simply relying on the pre-computed image connectivity during the design stage. Therefore, the matching of correspondences between the image pairs will be guided rather than applying a full pairwise matching where the probability of mismatching is much higher. Moreover, the information about the exterior orientation of the images and the object rough model will be utilised to limit the application of SIFT matching to small windows. This is expected to give more reliable tie point matching and to avoid the effect of repetitive structures (Fig.3.3).

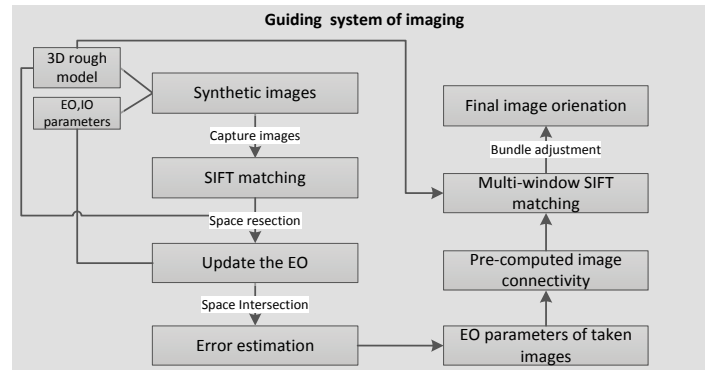


Fig.3.3: General methodology of the proposed guiding system.

3.4 Detection and recovery of gaps in the 3D point cloud

Dense point clouds either produced from range-based or image-based techniques are subject to the incompleteness or gaps. Gaps may have different causes like the effect occlusions, texture-less objects, or insufficient data collection or the bad design. For image-based point cloud, the dense matching technique of the oriented images as discussed in chapter 2 will result in a dense point cloud of the object. However, gaps can be found in the point cloud and should be detected and recovered to reach a complete 3D model.

In this thesis in chapter 6, a gap detection approach will be presented. The proposed methodology of detection is built on either a 2D space or 3D space representation. The 3D approach is based on a voxel representation of the whole space spanned by the point cloud. The empty voxels are to be filtered from blunders and then to check their visibility from the captured images. The partly occluded voxels will be considered as potential gaps in the point cloud. The final decision of detecting gaps will be assisted by checking the status of the surrounding or neighbouring voxels for every partly occluded voxel.

Based on the detected gaps, additional images are to be captured to improve the lack of coverage and to have a complete 3D model. Fig.3.4 illustrates the proposed methodology of this gap recovery objective.

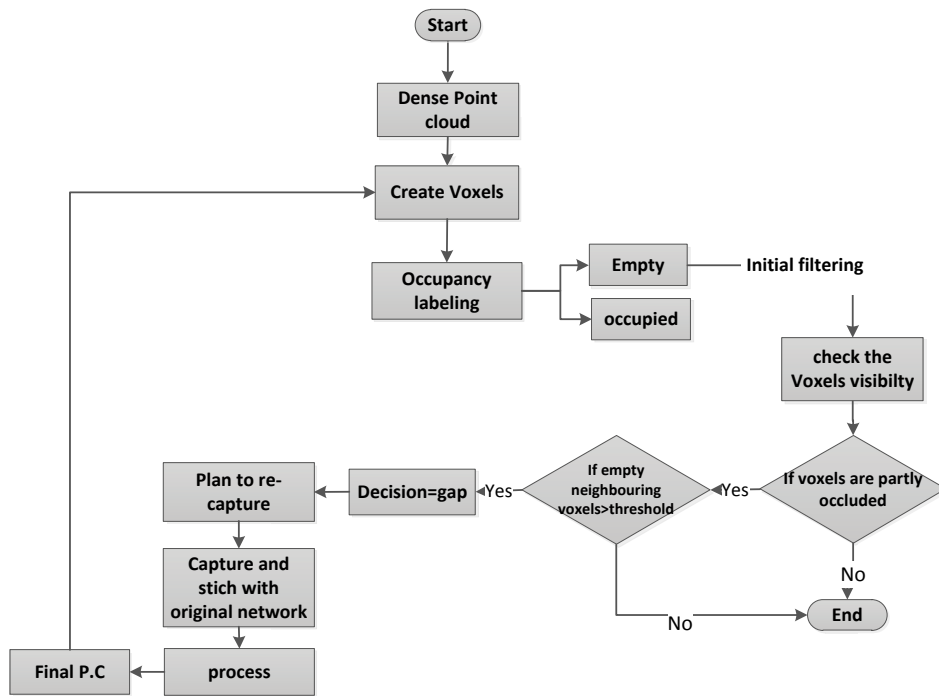


Fig.3.4: The gap detection methodology.

Chapter 4 Camera Network Design

According to the demands of portability, low cost, fast acquisition, and accuracy, images represent a suitable source to create the reality 3D models as explained in the previous chapters.

Although automatic techniques for measuring image coordinates and reconstructing camera orientations are achievable nowadays, automated creation of accurate and highly detailed 3D models still needs a well-designed camera network. Camera network design requires many efforts in planning and a high degree of proficiency (Remondino and El-Hakim 2006; Barazzetti et al. 2010), which is a costly operation, and it is mainly concerned with finding the suitable placements for the minimum number of imaging cameras.

The term 'optimal imaging configuration' within the camera network design is referring to have a camera network yielding a high accuracy of the reconstructed 3D scene points and complete coverage. This optimal network is constrained by having good imaging conditions and appropriate hardware. As presented in chapter 2, the optimization of networks has been developed and studied extensively by the geodesists. They, classified the problem into four classes (zero, first, second and third order design) basing on the requirement and use of the geodetic network. This classification of network optimization is identical with camera network design in photogrammetry. The optimal camera network means to find the appropriate placement of cameras and the best observation plan (number of images). This is quite demanded and applicable for industrial applications and deformation monitoring in close range photogrammetry (Fryer et al. 2007). Since these photogrammetric applications ask for a high positional accuracy, a wide baseline camera network is usually designed. Then, image measurements are applied either by manual interaction or using coded targets as control points. However, for 3D reconstruction with markerless image points and in contrast to the mentioned applications a short baseline network is needed as illustrated in chapter 1.

The objective in this chapter is to research about finding the optimal camera network for a highly detailed 3D modelling. As will be described in chapter 5, this should enable a non-professional to take the planned imagery suitable for the 3D modelling task. The developed method will be applied to the documentation of cultural heritage objects like buildings and statues. The method is based on finding a rough point cloud as obtained from the state-of-the-art SfM techniques. The optimal network will be acquired by finding at first, the sufficient number of imaging cameras for each object point and then optimizing the network accuracy. Fig. 4.1 illustrates the general methodology workflow for having the optimal camera network design of buildings.

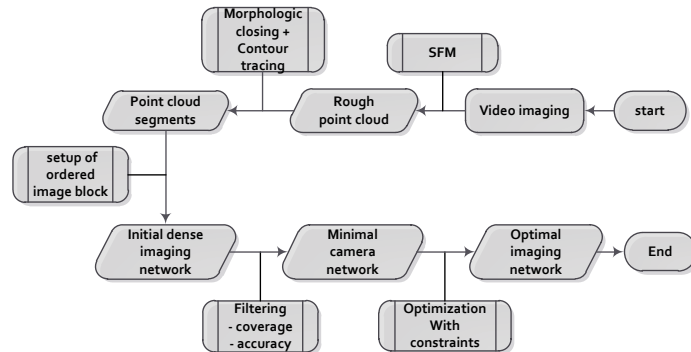


Fig. 4.1: The general methodology of camera network design.

4.1 Initial point cloud

Enough information about the study object like shape, size, and geometry necessitates the generation of primary or rough point cloud to proceed with the camera network design. This sparse point cloud is derived by applying the SfM technique using a video image stream of the object.

At this stage, the video imaging is to be used rather than the still shots for the reason of simplicity and the ability of non-professionals to do this mission. This means that the camera operator does not need to take care about the overlap percentage between the images and the places of taking the image shots.

Consequently, the SfM technique consists of matching features along the stereo images, which can be done, by feature tracking technique as mentioned earlier in chapter 1. One of the most popular standard algorithms for selecting and tracking feature points through an image sequence is KLT which is proposed by Tomasi and Kanade (1992) and further developed by Shi and Tomasi (1994).

In Fig. 4.2, different examples are shown where a camera tracking software like Boujou (Vicon 2010) is used. The video streams around a statue and a facade are processed by SfM and finally get these sparse point clouds.

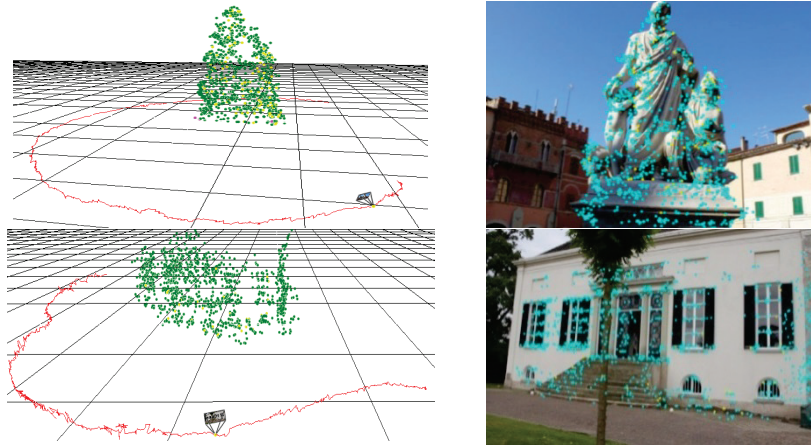


Fig. 4.2: Initial point clouds and video track.

4.2 *The subdivision of the point cloud and camera domain*

It is necessary when designing the imaging network of buildings to define the camera domain that is sufficient for having a complete 3D model. The implementation of this task starts by automatically decompose the point cloud of the building into facade planes (Fig. 4.3a). Therefore, the detection of individual facades will simplify the design by building a camera network for every detected façade. Then, all the individual networks will be connected by adding connecting cameras as described in section 4.3. This procedure will guarantee both, the full coverage of the object and the connection of adjacent facade camera networks to have ultimately a complete 3D model of the building. This subdivision methodology is more efficient for buildings than planning a ring camera network when U-shape facades exist. The surrounding ring network might not be convenient to cover the object at that U-shape part as shown in Fig. 4.3b.

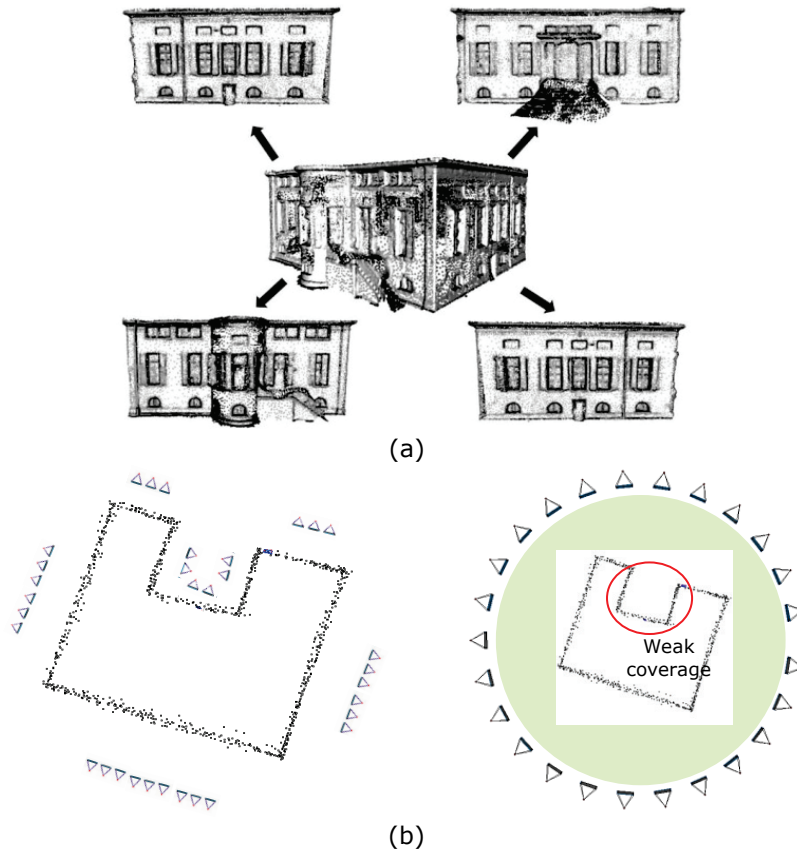


Fig. 4.3: (a) Subdivide the building point cloud into its facades. (b) Two types of camera networks for a U-shaped building with the subdivision into facades and the inefficient ring network design.

Accordingly, the following generic strategy for automatic camera domain identification will be followed (Fig. 4.4):

- Identifying the imaging location (outside or inside).
- Simplifying the object into adjacent segments: this task is based on the assumption of a 2D problem where the point cloud is projected into the 2D plane. Morphological closing is applied, and the building outline is traced and segmented into straight-line parts which correspond to facades. It must be mentioned that the assumption of a 2D problem is based on the fact of having a rough point cloud acquired for a street level video imaging. This video imaging will certainly produce points only on the facades rather than points on the roofs. However, the image dilation is expected to erase points on roofs as their density is relatively low.
- Subdividing the point cloud, according to the detected segments: this task is formulated in 3D where the points that fall inside the domain of a detected segment are considered as a single building facade.

Consequently, the initial camera network can be designed and placed for each detected segment (facade), in the right place that guarantees the full coverage of the object.

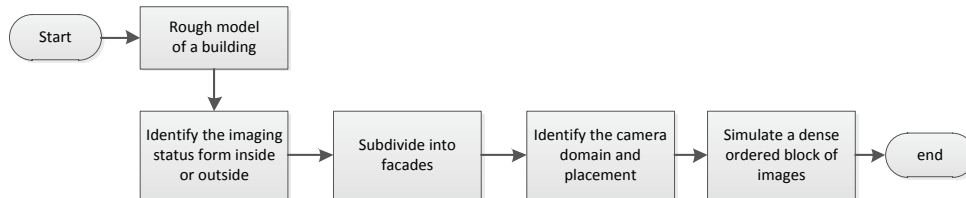


Fig. 4.4: Workflow of finding the camera domain of a rough cloud of a building.

4.2.1 The identification of imaging location

To build an automated dense camera network system of section 4.3 with a correct camera orientation, there is a need to identify if the cameras will be placed from outside or inside the object. In both cases, the developed algorithm starts by computing the best-fit circle for both of the initial video camera locations and the derived rough point cloud. The symbols 'r' and 'R' denote the radiuses of these two best fit circles respectively as shown in Fig. 4.5.

The algorithm output is to indicate either the designed cameras will be placed inside or outside as follows:

- If $R > r$ then the imaging is from inside.
- If $R < r$ then the imaging is from outside.

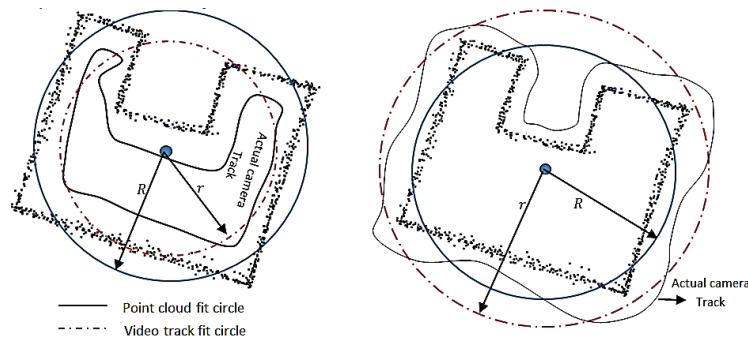


Fig. 4.5: Identification of cameras to be placed inside or outside the building.
(a) Inward camera domain ($R > r$). (b) Outward camera domain ($R < r$).

It must be noted that this is applicable for buildings; while statues are always imaged from outside and the camera domain can be represented simply as a circular network domain as will be shown in Fig. 4.7.

4.2.2 The camera domain placement

After the identification of the imaging and if the point cloud are properly segmented (section 4.2.3), the camera domains will be identified. This placement is based on computing the azimuth angles of the 2D segments in a clockwise direction (arrow head in Fig. 4.6). Therefore, if the detected object according to the object identification is to be imaged from outside, the camera domain is constructed to the left side by subtracting 90° from the azimuth direction while adding 90° if the imaging is from inside as illustrated in Fig. 4.6.

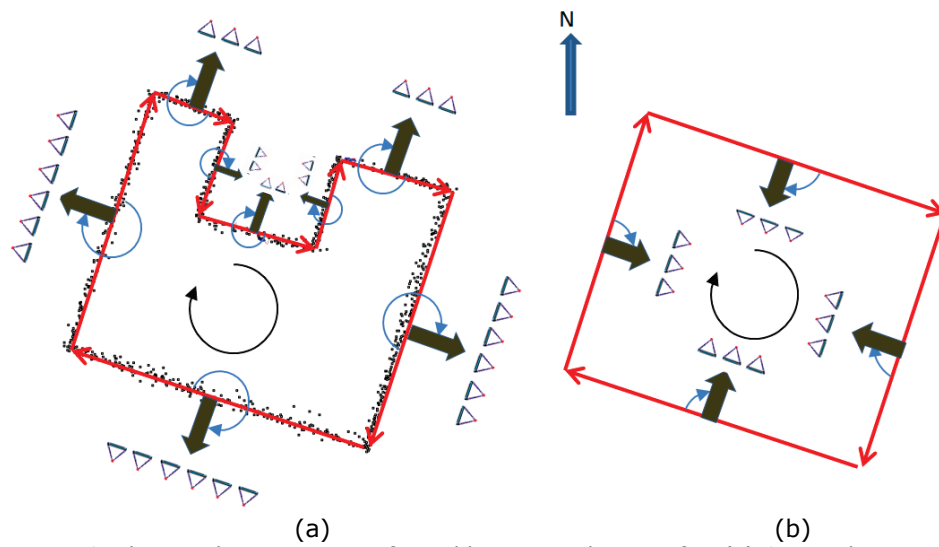


Fig. 4.6: The angular orientation of suitable camera domains for. (a) Outward camera domain object. (b) Inward camera domain.

However, for free form objects like statues and monuments, the camera domain can be adopted in most of the cases as a circular network around the statue as shown in Fig. 4.7.

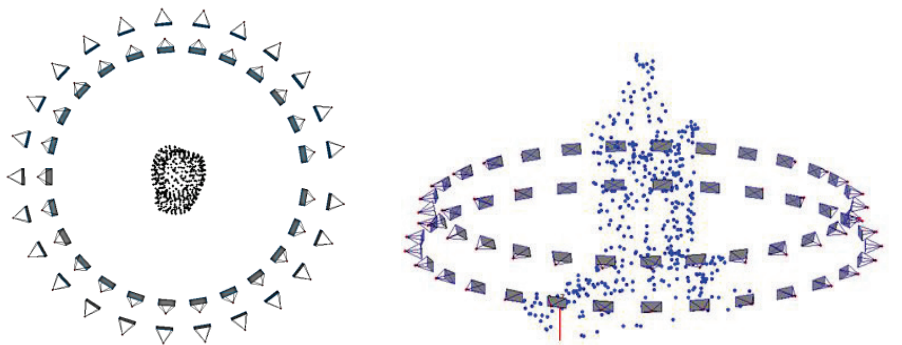


Fig. 4.7: The camera circular domain for statues and monuments.

4.2.3 The subdivision into facades

The identification of whether to place the cameras outside or inside is not enough for the completion of the network design. There is a need to specify their locations with respect to the building facades as stated previously to guarantee the sufficient coverage. Therefore, it is necessary to detect automatically, the object geometric primitives, like the planes of the building facades. The method is based on detecting facades and then to design a camera network for every façade (Fig.4.6). However, the adjacent camera networks should be connected by adding connecting cameras in a similar way introduced by Mason (1995) as described in section 4.4.

The subdivision methodology is illustrated in Fig. 4.8 by transforming the 2D point cloud into raster format, namely a binary image (BW). The size of this created BW image is chosen to guarantee a reasonable pixel dimension of 25cm to represent any real gap in the point cloud.

Accordingly, mathematical morphology techniques, namely, dilation, erosion, and contour tracing are used to extract the linear segments and the corner points of the building outlines. This is followed by removing redundant corner points along every segment to get the correct segments corner points that eventually define the building outline.

Finally, the point cloud will be divided into sub clouds relating to each detected segment. These facade rough point clouds will be used for designing the camera network of section 4.3.

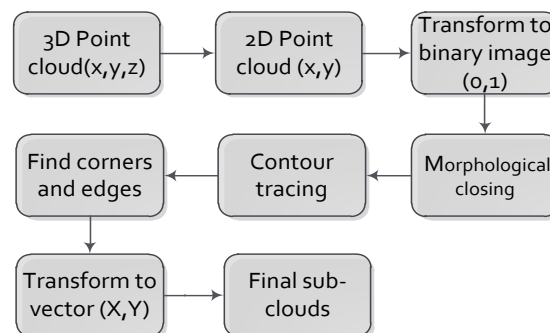


Fig. 4.8: Point cloud subdivision methodology.

The image dilation will be applied in order to guarantee a continuous object boundary in case of any existing gap in the raster image. The dilation will be followed by erosion (morphological closing) using a square structural element SE. The contour-tracing algorithm is applied in order to produce an ordered list of the outline or a contour of the building boundaries.

Once the contour of a given shape is extracted, corner points or edges endings can be extracted. Therefore, the correct extraction of the contour will produce more accurate features, which will increase the chances of correctly classifying a given pattern (Ghuneim 2010). Ultimately, the normals of the traced pixels are computed to extract or refine the detected edges or corners. (Fig. 4.9).

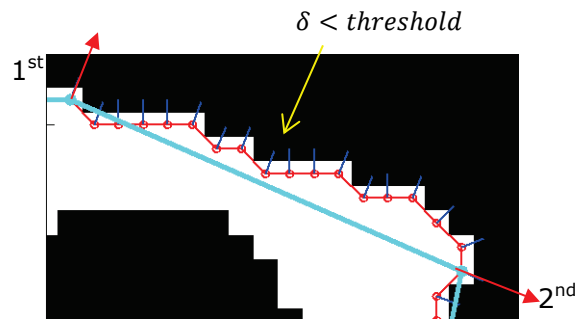


Fig. 4.9: Edges extraction by testing the normal.

The final step is to divide the point cloud into sub-clouds relating to each detected segment or building facade. This is accomplished by searching for the points that are within the domain bounded by the minimum and maximum coordinate limits of each detected facade segment. Fig.4.10 illustrates the whole subdivision procedure.

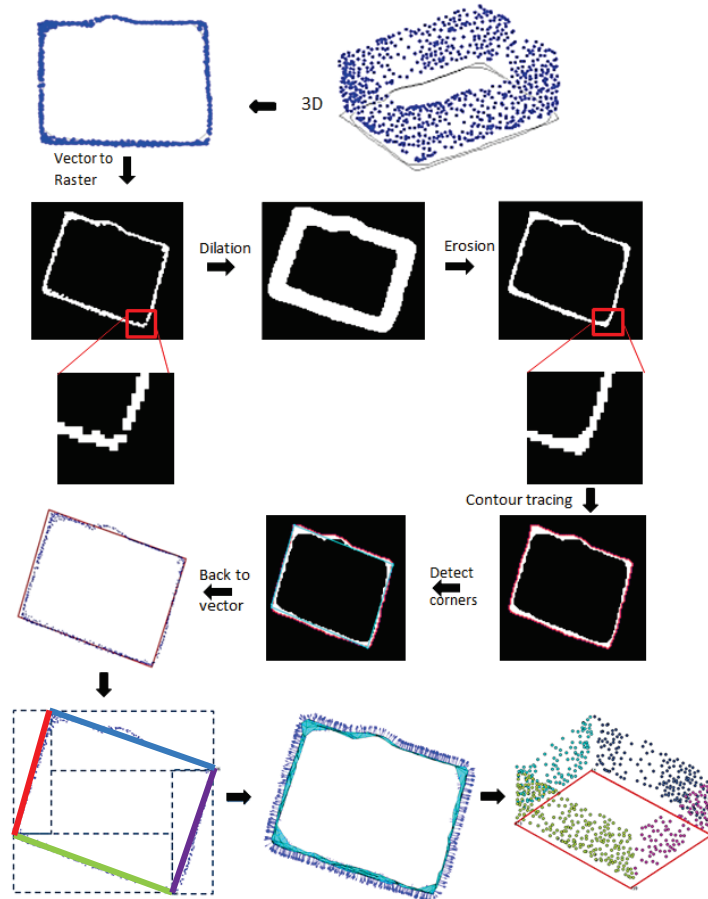


Fig. 4.10: Point cloud subdivision technique.

However, the adjacent facades of buildings are not always right angled and cause the bounded domains to interfere. This problem is to be avoided by testing the triangulated surface normal directions of the point cloud against the normal azimuth direction of the detected segment (Fig. 4.11).

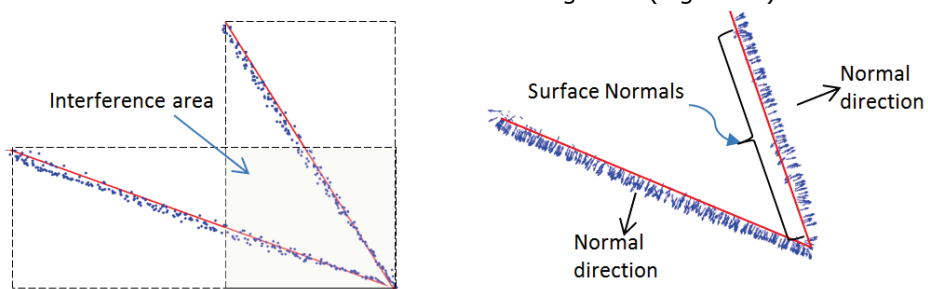


Fig 4.11: Surface normals are used to avoid interference of sub-clouds.

It should be mentioned that for curved facades, the proposed method results in multiple connected segments. The following section 4.3 is describing the procedure for designing and computing the dense initial camera network.

4.3 Initial camera network design in blocks

The imaging network design is usually formed in a block shape where overlapped images are designed in multiple overlapped strips. In close range photogrammetry, a block can be designed either in an ordered image sequence or unordered sets of images. The first design has a linear computational cost with respect to the number of images. For unordered sets of images, it is necessary to check all possible image pair combinations to determine the ones sharing sufficient correspondences (Barazzetti et al. 2010) Therefore, each image must be compared with all the others, leading to a high computational cost. For this reason, we used the first type of an ordered imaging block as shown in Fig. 4.12.

In the design step there are two possible choices, either to start from the maximum observation scheme and minimize until finding the optimal or to start from the minimum observations and maximize until reaching the optimal observation plan. In this thesis, the first choice is adopted by designing a dense camera network with an overlap percentage of (80-90%). This strategy of maximum to minimum is especially suitable for finding the minimal number of images among a huge number of captured images for an object like the internet collection images (Snavely 2008).

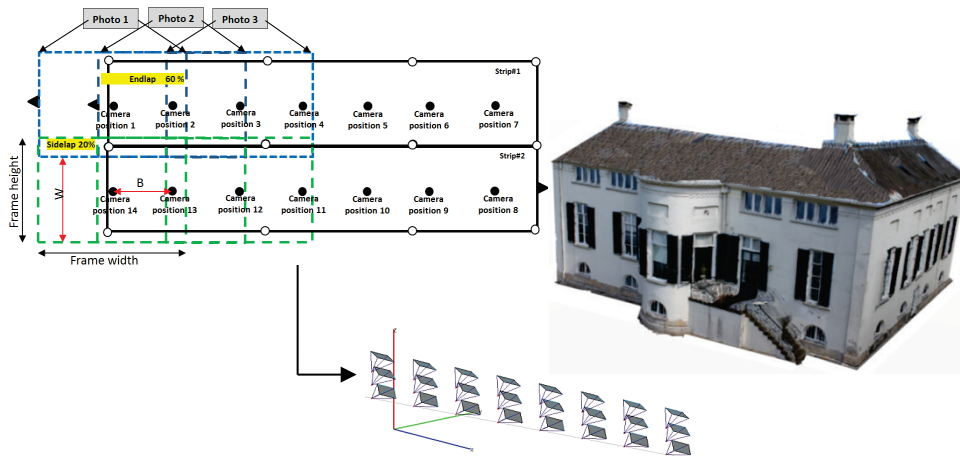


Fig. 4.12: Photo layout for standard ordered-block in terrestrial photogrammetry.

The algorithm that is used to design the imaging blocks is based on the conventional aerial block design as follows (Wolf and DeWitt 2000):

$$W = scale * ((1 - sidelap) * fh) \quad (4.1)$$

$$B = scale * ((1 - endlap) * fw) \quad (4.2)$$

$$strip = \left(\frac{Z_2 - Z_1}{W} \right) + 1 \quad (4.3)$$

$$photos = \left(\frac{Length}{B} \right) + 1 \quad (4.4)$$

Where

- W : The lateral advance per strip
- B : The imaging base
- $strip$: The number of strips per block
- $photos$: The number of photos per strip
- fh : The camera frame height
- fw : The camera frame width
- $Length$: The object length
- $Scale$: Image scale number
- Z_1, Z_2 : The upper and lower height of the point cloud

Since the heights that are accessible by the camera operator are chosen to be in the range of (0.5 – 2 meter), we will have a block design of two strips of images in most of the cases to completely capture the object of a cultural heritage building in stereo as will be shown in the tests.

The viewing angle ω of the cameras toward the object will be computed for each strip according to the difference in height between the object and the camera as shown in Fig. 4.13.

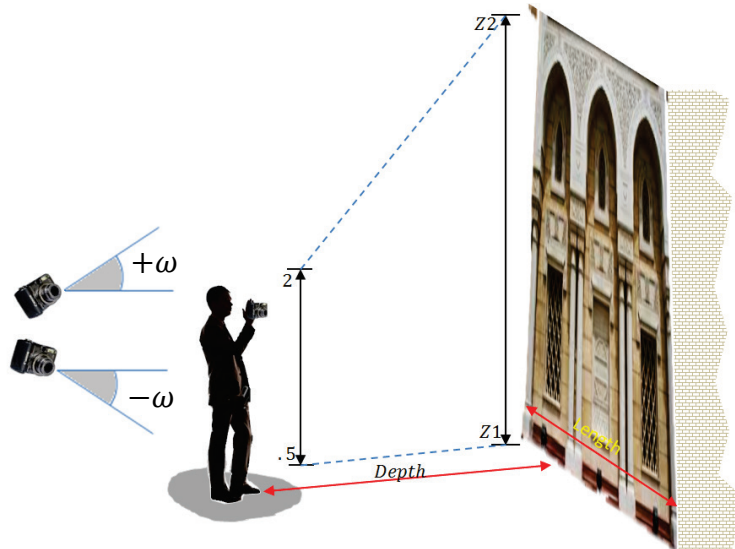


Fig. 4.13: The relation between the camera accessible heights and object height.

The distance between the camera and the study object (*Depth*) is an important parameter in the camera placement objectives and this is influenced by many factors like camera field of view, user-defined resolution, positional accuracy, depth of field and number of imaged points (Saadatseresht et al. 2005).

The required object minimum resolution or *ground sample distance* (GSD) is an important factor in determining the maximum depth distance between the camera and the object. The pixel size (*pixel*) can be factored beside the object resolution to find the depth distance as follows:

$$Depth = \frac{GSD * f}{pixel} \quad (4.5)$$

On the other hand and according to the accuracy requirements, the depth can be determined within the scale ($f/Depth$) as follows (Fraser 1989; Luhmann T. et al. 2006).

$$\sigma_{XY} = (q * scale * \sigma_{image}) / \sqrt{K} \quad (4.6)$$

Where σ_{XY} is the standard error in the object point, q is a design factor expressing the strength of the basic camera station configuration (0.4-1.1), σ_{image} is the standard error in the image coordinates and K is the number of images per camera station.

Nevertheless, in the camera network design, the minimum depth will be selected as the designing distance to maintain both demands of resolution and accuracy.

After setting all the requirements of the imaging block design and computing the segmented point cloud, the camera exterior orientation parameters are defined.

Fig. 4.14 illustrates the relationship between the simulated camera system (x, y, z) and object coordinate system (X, Y, Z). The designed network is based on the rough point cloud created from the video imaging as mentioned in section 4.1. However, the initial camera network will be reoriented during the optimization computations of section 4.7 to accommodate for the object points with a better coverage and accuracy.

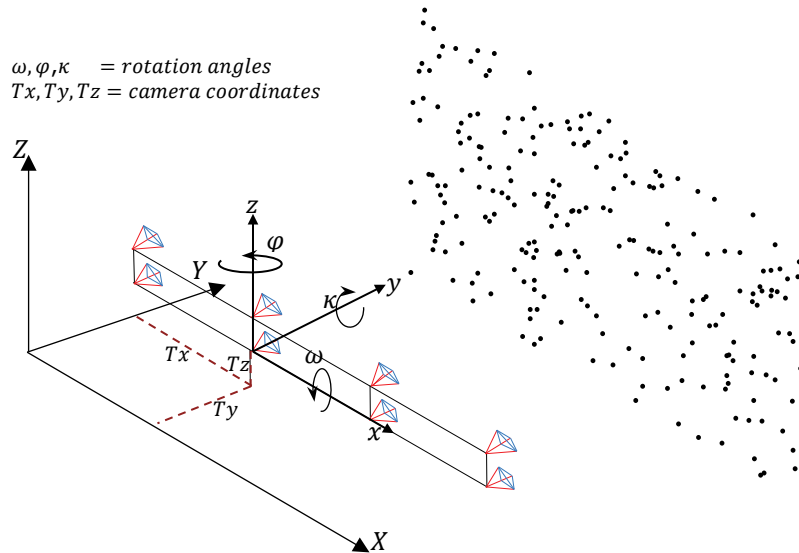


Fig. 4.14: Camera and object coordinate systems.

Moreover, the transformation between the object system and the imaging system needs to define the initial interior orientation parameters of the imaging camera like the focal length, image format, and the pixel size. However, for the image orientation, knowing the calibrated interior parameters is crucial to have accurate results. These parameters can be determined either earlier before use by lab camera calibration procedures or computationally during the image orientation by bundle adjustment (self-calibration). For the planning, however, some approximate values would be good enough.

4.4 Connecting cameras

To ascertain the complete connected and strong geometric camera network, it is important to add connecting cameras between the sub point clouds of the detected facades (Fig. 4.15). This is also to avoid the wide baseline imaging between the cameras that belong to adjacent facades and to add cameras that are viewing points belong to two facades simultaneously. The connecting cameras will be designed in a similar way to the facade cameras of section 4.3.

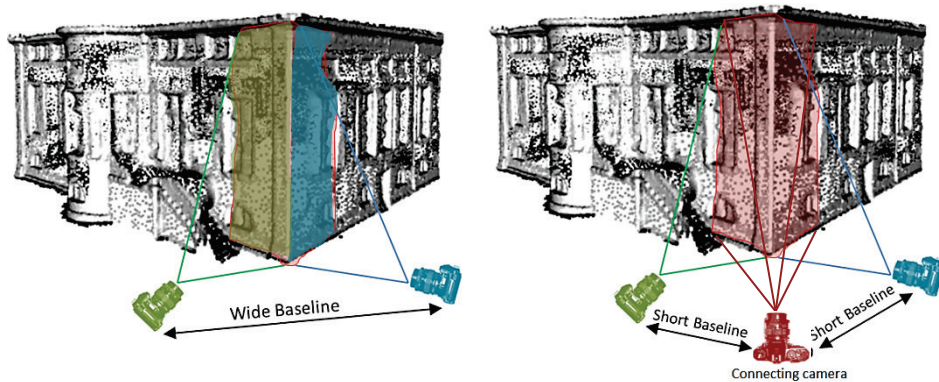


Fig. 4.15: Connecting camera placement around a building and its importance for 3D modelling.

In this thesis, we add three cameras for this task of connecting, however, we can place more cameras to avoid any shortage in the imaging coverage.

4.5 Minimal camera network

The optimal camera network design as mentioned earlier in chapter 1 is to have the minimum number of cameras in an optimal configuration which guarantees the sufficient coverage and accuracy, as pre-defined by the user. The minimal camera network is expected to produce the same object point cloud given by a dense network. In this sense, the minimal network is suitable to save the processing time and computer memory during the image orientation and dense matching especially when using high resolution images. Many application examples are asking to have the minimal networks like the disaster mapping and rescue operations or for the documentation of some precious cultural heritage objects in museums without disturbing the visitors (Alsadik et al. 2014a).

The strategy of finding the minimum number of cameras is based on designing a dense imaging block as discussed earlier in section 4.3 and then trying to filter out the cameras that are redundant in the sense of coverage and accuracy. Finally, optimization can be applied to modify and adjust the camera orientation and placement to minimize the expected errors in the point cloud.

In this thesis, the minimal camera network can be computed by using one of the proposed two different strategies as follows:

1. Minimal network by filtering for coverage requirements.
2. Minimal network by filtering for accuracy requirements.

Every suggested technique of filtering the dense network has advantages and disadvantages related to: the final accuracy of the point cloud, the percentage of coverage, and the required time for processing. The minimal camera network role in the general workflow of the camera network design is shown in Fig. 4.16.

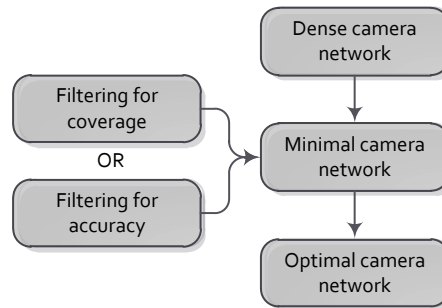


Fig. 4.16: The minimal camera network in the general workflow of design.

4.5.1 A minimal camera network based on filtering for coverage

The motivation of using this technique for filtering a dense block of images is based on the concept of having at least three cameras viewing simultaneously every object point. Therefore, the cameras are considered redundant if they only result in coverage by more than three cameras as shown in Fig. 4.17.

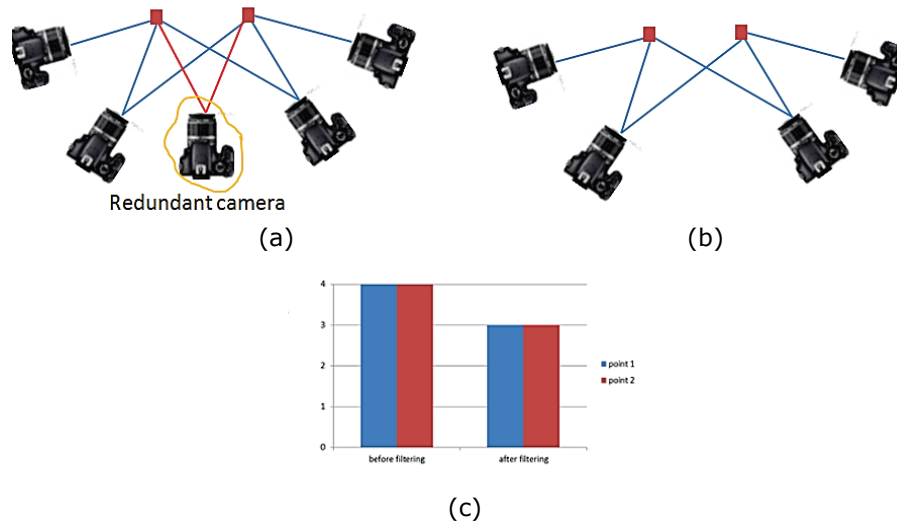


Fig. 4.17: The concept of filtering redundant camera. (a) Before filtering. (b) After filtering. (c) Number of covering cameras before and after the filtering.

The methodology of filtering is summarized as follows and shown in Fig. 4.18:

- 1- Divide the derived rough point cloud of the object into over-covered and fair-covered. Over-covered points, are the points that appear in more than three cameras while fair-covered points, refer to the points that appear in three cameras.
- 2- Label the cameras as redundant or significant based on the appearance of the over-covered points and fair covered points.
- 3- Arrange the redundant cameras involved in imaging over-covered points according to their coverage (number of points) in an ascending order. The reason for this arrangement is to cancel the redundant cameras that are imaging a fewer number of points and to keep the other cameras.
- 4- Check the effect of the camera cancelation on the B/D ratio. Accordingly, cancel the camera that is involved in imaging only the over-covered point group and doesn't produce a large B/D ratio in the network configuration.
- 5- Test the filtering iteratively according to the computed coverage after the camera cancelation and re-label the point cloud in each iteration into over-covered and fair-covered.
- 6- The same procedure of step 3 is followed, and a new camera is cancelled. The filtering is repeated until no more redundant cameras involved in imaging only over-covered points.

4.5.2 A minimal camera network based on the accuracy of the object points

The motivation of using the filtering for point accuracy is based on the prescribed relation between the ray intersection geometry or B/D ratio and its effect on the predicted precision at the intersection point $(\sigma_x, \sigma_y, \sigma_z)$ as shown previously in Fig. 1.1. Therefore, the technique is based on considering the cameras that have the least impact on the predicted accuracy as redundant.

Computationally, the covariance matrix of the unknown points Q_s is to be evaluated (Appendix A) and the effect of each camera on this error is to be computed.

The least effective redundant camera in terms of accuracy will be neglected. The whole procedure of filtering will be iterated until the removal of further imagery would lead to a violation of the accuracy requirement.

The filtering procedure based on accuracy is partly similar to the previous filtering method of section 4.5.1. The summarized procedure is:

- 1- The cameras involved in imaging over-covered points are classified as redundant cameras and is subject to filtering out according to the accuracy requirements.
- 2- Filter out the redundant cameras based on accuracy by computing the covariance matrix Q_s of every point. The redundant camera with the least impact on the total accuracy will be filtered out.

The workflow of both described filtering techniques of coverage and accuracy requirements is shown in Fig. 4.18.

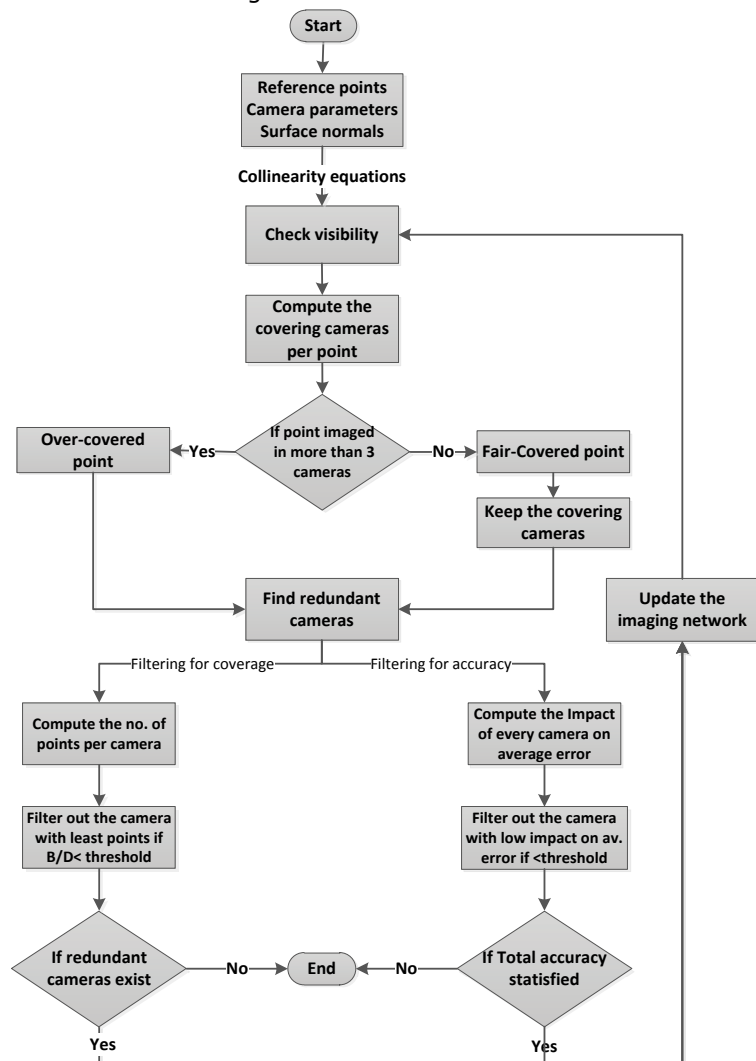


Fig. 4.18: The workflow diagram of filtering for coverage or accuracy requirements.

4.6 Visibility requirement

The sparse point cloud which is generated automatically by the SfM technique is to be tested for visibility from the designed camera locations. This visibility testing is very crucial for having a realistic and efficient camera network in terms of coverage. As explained in chapter 2, there are different techniques that can be used for this purpose. For simplicity and because of the sparse point cloud at this stage, the preferred methods are either the test of the triangular faces normal or to use the HPR method. The effect of using the visibility testing to design the camera network is simply shown in Fig. 4.19 in a simulated building. After filtering, four cameras are needed to have fully visible corners of the building by testing the visibility. However, only three cameras are found if the visibility condition is not considered which is incorrect.

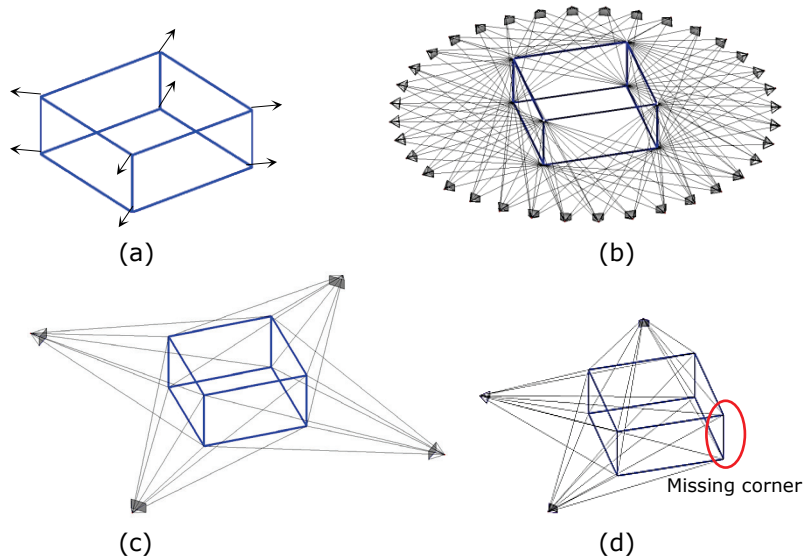


Fig. 4.19: The effect of visibility analysis on the design. (a) A simple building framework with normals. (b) The dense network configuration. (c) Minimal network with visibility test. (d) Minimal network without visibility test.

4.7 Optimization technique

As mentioned earlier in this chapter, if the predefined accuracy of the project is not reachable after computing the minimal camera network, an optimization computation of the camera network will be applied. The mathematical optimization overview is discussed in Appendix A. The modelling of the camera optimization can be summarized in Fig. 4.20.

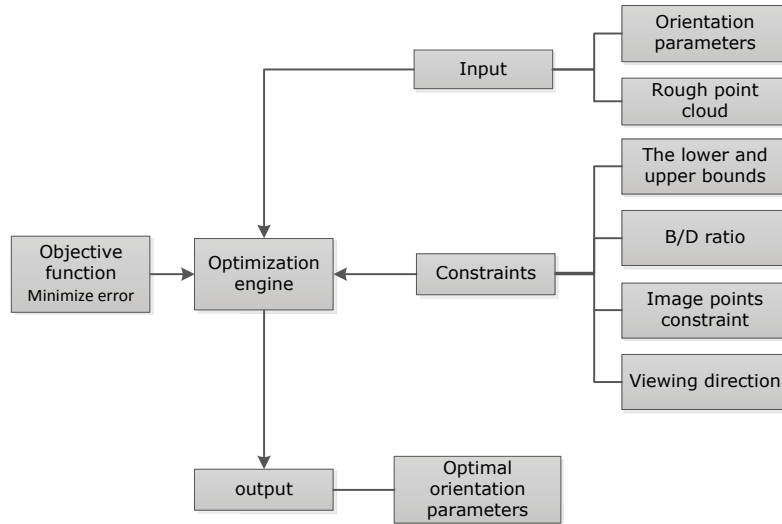


Fig.4.20: The workflow of the optimization computations.

The developed optimization objective function as shown in Fig. 4.20 will run to minimize the total error in the object points (equation 4.7). However, there are different constraints that must be satisfied during the optimization as will be shown in section 4.7.2.

Generally, the optimized network must be constrained according to the allowed B/D ratio, which can guarantee the effective dense reconstruction and accurate ray intersection. The adopted B/D ratio is 15% - 30% according to Hullo et al. (2009) and the 3 by 3 rules of CIPA (Waldhaeusl and Ogleby 1994). This ratio is chosen to have a compromise between the higher accuracy and the efficient dense matching for the 3D reconstruction.

The derived B/D ratio is also related to the allowable distance between the camera and the object as mentioned previously in section 4.3 where the depth is a function of the predefined GSD, scale, camera angular field of view and accuracy. Moreover, the optimization will compute the camera orientation in a way that satisfies the uniform distribution of the image points around the image centre as possible. This is supposed to improve the final accuracy since more points will appear in the optimal images. However, keeping the same points after filtering to minimal network to remain visible in the same cameras is necessary to preserve the pre-designed frequency of at least three cameras per image point.

4.7.1 The formulation of the camera network optimization problem

The optimization problem with camera placement needs to define precisely the input and output parameters, which are necessary to formulate the objective function $f(x)$ that can guarantee the final goal of optimal camera network a shown in the workflow of Fig.4.20. The input and output parameters can be listed as follows:

1- The input data and parameters:

- Point cloud $(X_j, Y_j, Z_j, j = 1:n)$ coordinates acquired from video stream tracking related to each cluster.
- The initial exterior orientation parameters for each possible (designed) camera $(\omega_i^o, \varphi_i^o, \kappa_i^o, Tx_i^o, Ty_i^o, Tz_i^o) \quad i = 1:m$. These exterior orientation elements will be collected in one vector (x_0) which represents the initial guess of unknowns for the next optimization iteration step.

2- The output parameters:

- The optimal exterior orientation parameters $\omega_i, \varphi_i, \kappa_i, Tx_i, Ty_i, Tz_i$ for each camera i in the whole network.

As mentioned previously, the mathematical model is represented by collinearity equations, and can be linearized by the first-order development of Taylor series. Moreover, with the redundant observations, least square method is usually followed to compute the adjusted parameters (appendix A).

The designed optimization objective of increasing the accuracy is to achieve an optimal form of Q_s which is the covariance matrix of object point coordinates (X, Y, Z) . Consequently, this will guarantee the maximum accuracy (Fraser 1989) with the optimal minimum number of cameras. This objective function is expressed in equation (4.7) by minimizing the computed average error in $(X, Y, \text{and } Z)$ for each object within n unknown object points, which is computed according to the collinearity equations model.

$$f(x) = \min(\sum_{i=1}^n Q_s^i / n) \quad (4.7)$$

This is meant to improve the accuracy of the global project, and thus, the number of visible points in every image increases after the modification in the image orientation.

At this stage, the input parameters are defined in the next step of optimization. The most important and costly computational step is to solve

the mathematical optimization problem. This is due to the huge amount of input data and parameters to be processed during the computations. The camera optimization problem is a nonlinear problem and needs to be constrained to obtain real results that satisfy the final goal for the 3D modelling of the object. The next subsection 4.7.2 will explain the necessary constraints involved in the camera network optimization for 3D modelling.

4.7.2 The constraints on optimum camera placement

The camera placement problem is influenced by many design and geometric constraints, which can be listed as follows:

- The lower and upper bounds of the estimated parameters for each designed camera

$$\left. \begin{array}{l} -25^\circ < \omega_i < 25^\circ \\ -25^\circ < \varphi_i < 25^\circ \\ \kappa_i = 0 \text{ or } \kappa_i = 90^\circ \\ T_{xi} - \delta x < T_{xi} < T_{xi} + \delta x \\ T_{yi} - \delta y < T_{yi} < T_{yi} + \delta y \\ 0.5m < T_{zi} < 2m \end{array} \right\} \quad (4.8)$$

The allowed shifts in the camera position δx and δy depend on the available workspace that can be occupied around the object. The allowed camera height is chosen as stated earlier in section 4.3 to fit a layman. For angles ω and φ the allowed change is designed within 50° while κ is designed to keep the camera either in a portrait or landscape pose. The bounding limits are illustrated in Fig. 4.21 clearly where the initial orientation is red coloured.

- The distance between the camera and the object:

$$Min_{depth} < Depth < Max_{depth}, \quad Depth = \sqrt{\Delta x_{ik}^2 + \Delta y_{ik}^2 + \Delta z_{ik}^2} \quad (4.9)$$

Where

Depth: The distance from the camera i to the cluster of points k .

Min_{depth}, Max_{depth} : The minimum and maximum allowed distance between the camera and the object. These values are chosen in most of our experiments in the range of 2 – 30 m respectively.

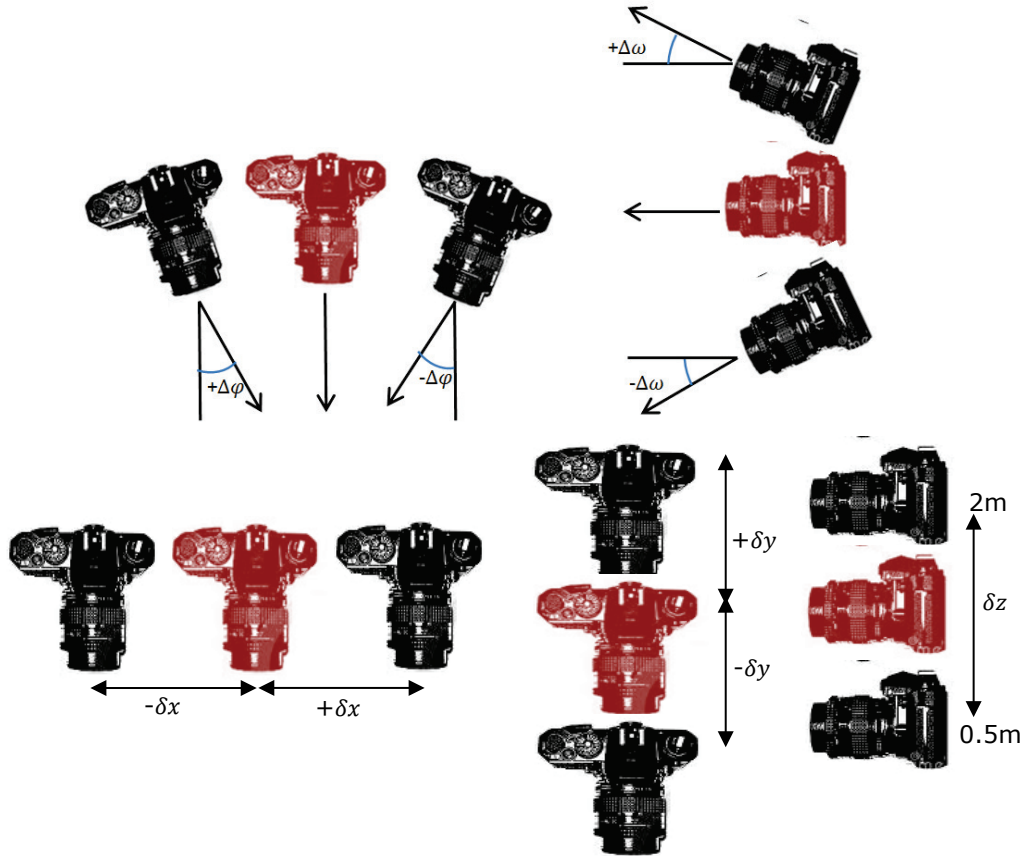


Fig. 4.21: The bounding limits of the optimal camera orientation.

- The distance between the designed cameras:

$$Min_B < B < Max_B, B = \sqrt{Dx_{ij}^2 + Dy_{ij}^2 + Dz_{ij}^2} \quad (4.10)$$

Where

B : The distance between camera i and j .

Min_B, Max_B : The minimum and maximum allowed distance between cameras, according to the predefined range of B/D ratio.

- Geometric design constraint:

This constraint is formulated by minimizing the absolute value of the angle between the facade mean normal and the designed camera optical axis as in equation (4.11).

$$\delta < threshold \quad where \quad \delta = |N_{dir} - Cam_{dir}| \quad (4.11)$$

N_{dir} = normal direction of the local point set

Cam_{dir} = the design camera axis direction

- The constraints of the image coordinate:

This means to constrain the average of the image coordinates \overline{xp} , \overline{yp} (in p.p. system) to equal zero. This constraint is supposed to modify the camera orientation to distribute the points evenly around the image centre.

$$\begin{aligned}\overline{xp} &= 0 \\ \overline{yp} &= 0\end{aligned}\tag{4.12}$$

- To keep the minimum of three viewing cameras per point as well designed before running optimization, another constraint is formulated. Every image point is constrained to remain observed from the same camera after optimization.

$$\begin{aligned}-w/2 < xp_i < w/2 \\ -h/2 < yp_i < h/2\end{aligned}\tag{4.13}$$

Where h and w represent the image format height and width respectively. These values are chosen in this thesis as for Canon 500D to be 14.8 mm and 22.3 mm, respectively.

- The constraint of covered area by image points

This constraint can be used to support the appearance of maximum number of image points in the optimal images beside their good distribution. This is done mathematically by computing the convex hull area of the points.

4.8 Summary of methodology

The algorithm of optimization is summarized as follows and the general diagram of the camera network design is shown in Fig. 4.22:

Input: A vector function x_0 of the initial parameters estimation, and n point cloud coordinates P .

Output: A vector function \hat{x} of the optimized camera parameters, minimizing $f(x) = \left(\frac{\text{trace } Q_s}{3}\right)$ for each point i .

Algorithm:

Call function optimcon: (x_0) optimal constraints

Call function optimcam: (x_0) objective function.

$[\hat{x}, fval] = \text{fmincon}(\text{optimcam}, x_0, lb, ub, \text{optimcon}, \text{options});$

end

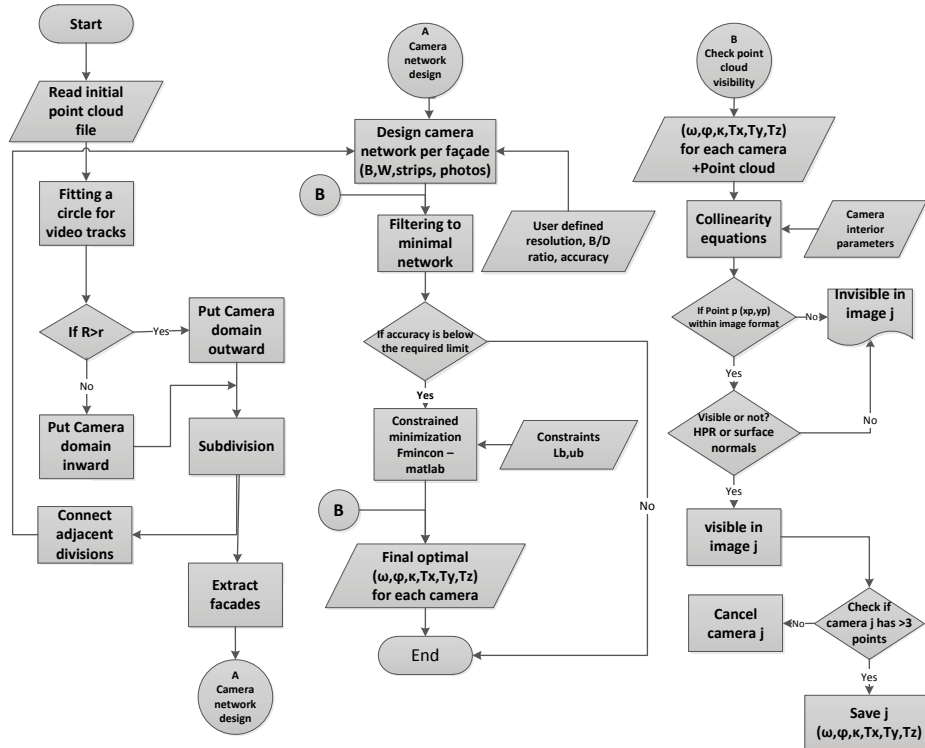


Fig. 4.22: The general workflow of the camera network design.

4.9 Experimental tests

The main aim of the experiments is to test the developed technique. Therefore, we implemented experiments on three different cases, a simulated network, a building, and a statue. The design is based on having a rough point cloud as described previously. These point clouds are scaled into ground truth dimensions during the derivation process; however, a local coordinate system is adopted in these cases. The imaging is simulated as mentioned earlier with Canon - 18mm with a $22.3 \times 14.8 \text{ mm}^2$ frame size and 4752×3168 pixels resolution. The image measurement precision of $10 \mu\text{m}$ is assumed to evaluate the accuracy. The designed B/D ratio is selected as small as 0.1 to create a dense imaging block for the building and the statue tests respectively. In the first test of a simulated point cloud, the two filtering techniques for coverage and accuracy will be investigated and compared to the optimal network. In the second test of a building in section 4.9.2 the filtering methods for coverage and for accuracy will be tested beside the optimal network. In section 4.9.3, a camera network is designed for a rough point cloud from a statue where the filtering for coverage is applied and then compared to the dense and the optimal network. The comparison between

the camera networks in the tests are based on evaluating the maximum and average errors in the point cloud beside the number of covering cameras.

4.9.1 Simulation test

A simulation of a dense imaging network is created where 22 cameras are viewing eight points as shown in Fig. 4.23. The overlap ratio is selected as 80% to create a dense camera network.

To evaluate the expected accuracy in the object points, the standard deviations $(\sigma_x, \sigma_y, \sigma_z)$, are estimated by image triangulation (image intersection). Therefore, a perturbation of normally distributed errors with a standard deviation of $10 \mu\text{m}$ is added to the image coordinates as mentioned earlier.

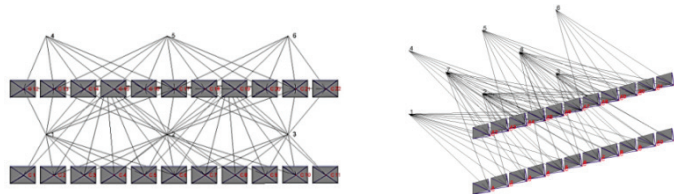


Fig. 4.23: A simulated dense camera network.

The ellipsoid of errors is evaluated as shown in Fig. 4.24a with an exaggerated scale of 100 to visualize the expected error of the object points. The optimal configuration of the imaging network is shown in Fig. 4.24b with the computed ellipsoid of errors.

The amount of coverage after optimization is improved significantly as shown in Fig. 4.24d. The optimization run is illustrated in Fig. 4.24c which shows the stability and convergence of the solution.

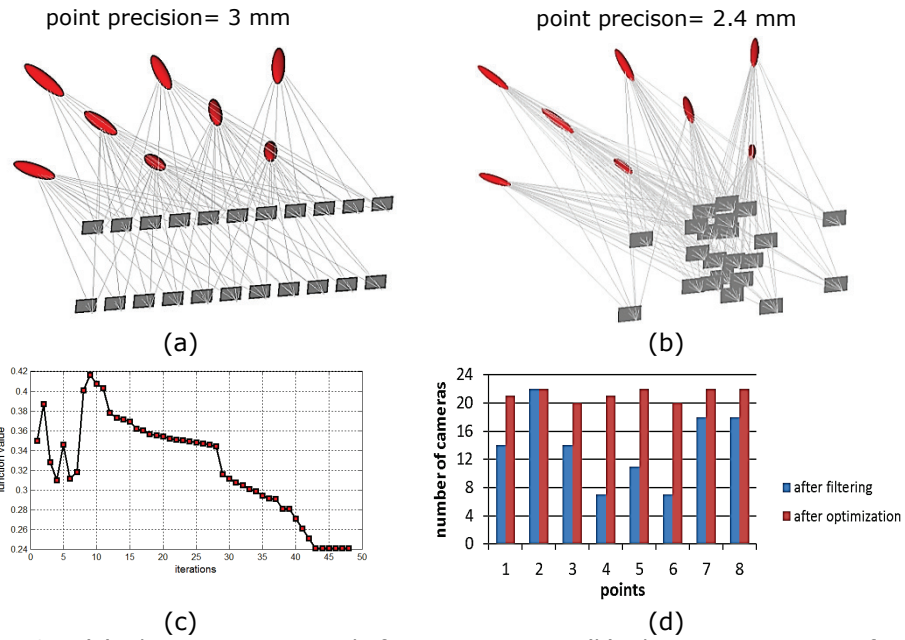


Fig. 4.24: (a) The point precision before optimization. (b) The point precision after optimization. (c) The convergence of solution of optimization. (d) The number of the covering cameras before and after the optimization.

The filtering techniques are also implemented to have the minimal camera network with respect to coverage and accuracy requirements of minimum three cameras. The first filtering is implemented for coverage as shown in Fig. 4.25a to preserve an average B/D ratio of 30%. Optimal network is shown in Fig. 4.25b and the run of the optimal solution is shown in Fig. 4.25c.

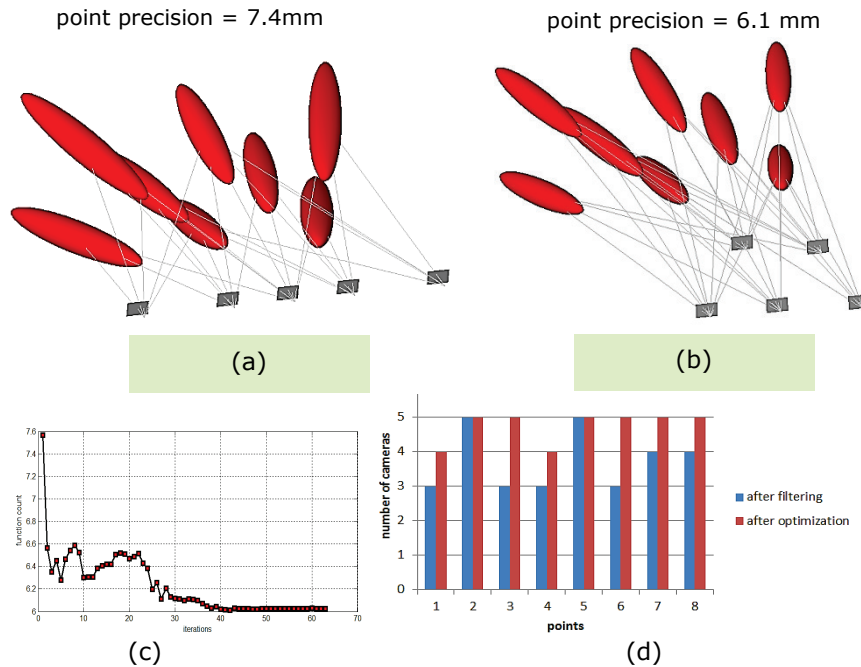


Fig. 4.25: The simulation of the filtered imaging network and results of filtering and optimization. (a) Filtered net for coverage. (b) Optimal net. (c) Optimization run plot. (d) The number of the covering cameras before and after the optimization.

The filtering for accuracy requirements is also applied to the dense network to preserve the average point precision to less than 10 mm. The filtering resulted in four cameras which are then optimized as shown in Fig. 4.26.

Adopting average point precision rather than the precision of the worst reconstructed point in this method is because of the restriction of camera altitude. As stated in section 4.3, the camera accessible height is restricted in the range of 0.5 – 2 meters of the camera operator. Therefore, worse points are usually found in the upper parts of objects where the ray's intersection is weak and the accessibility is limited. This implies to design the filtering to preserve the average point precision.

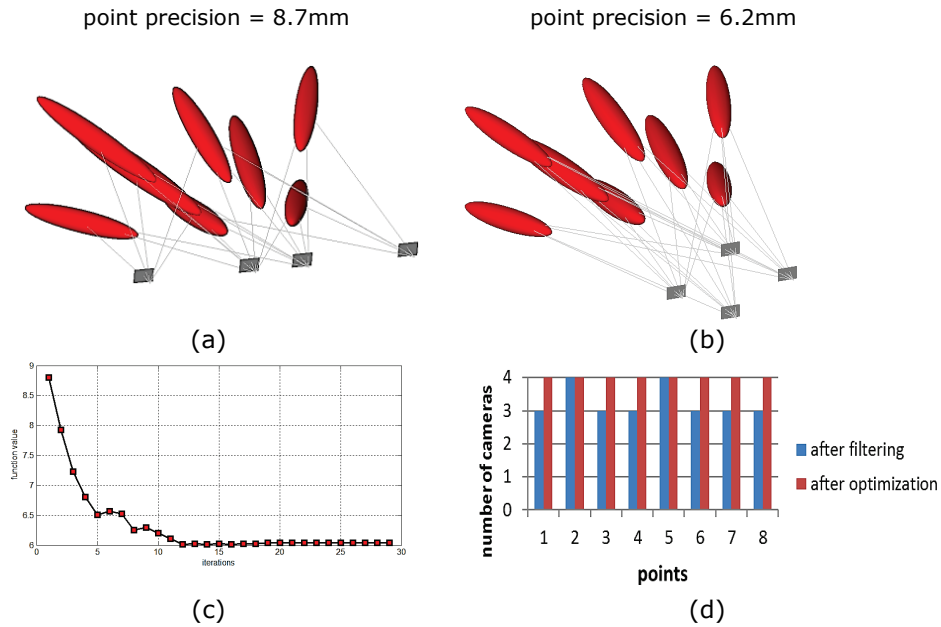


Fig. 4.26: (a) The point precision after the filtering for accuracy. (b) The point precision after optimization. (c) The convergence of solution of optimization. (d) The frequency of covering cameras before and after the optimization.

It must be mentioned that the optimizations stopped because the size of the current step is less than the default value of the step size tolerance and constraints are satisfied to within the default value of the constraint tolerance.

As a conclusion, the dense image network provided with the highest possible positioning accuracy of the object points. However, after optimization, the network in Fig. 4.24b provided with better positional accuracy. This seems logical since there are more redundant observations of image rays per point. The filtering approach based on coverage with five cameras gave slightly better accuracy than the filtered network based on accuracy with four cameras. The reason to have five cameras after using the coverage filtering method is based on satisfying an average B/D ratio of 30%. However, in terms of the minimum network that satisfies all the designing constraints, the accuracy filtered network is the optimally filtered network.

For optimal networks, the used method proved in all the cases the stability of the solution and convergence as shown in Figures 4.24c, 4.25c and 4.26c. Moreover, better accuracy and coverage are also gained after optimization of Figures 4.26d, 4.25d and 4.24d. It is worth to notice, that the errors in the viewing or depth direction are larger and this is logical since it is related to

the B/D ratio. A smaller error can be attained whenever a larger base between the cameras found as described in chapter 1.

4.9.2 Building\

A real point cloud has been extracted from a building where a video stream (Fig. 4.27) of 177 frames is captured around the building with a resolution of 1920*1080 pixels.



Fig. 4.27: Video stream for a building (sample images).

To create a rough point cloud, the images are processed by using SfM to get a sparse point cloud of 3320 points of the building as shown in Fig. 4.28a and the rough model of Fig. 4.28b. The rough point cloud is segmented by using the presented technique of cloud subdivision mentioned in section 4.2 and the results are summarized in Fig. 4.28c and Fig. 4.28d.

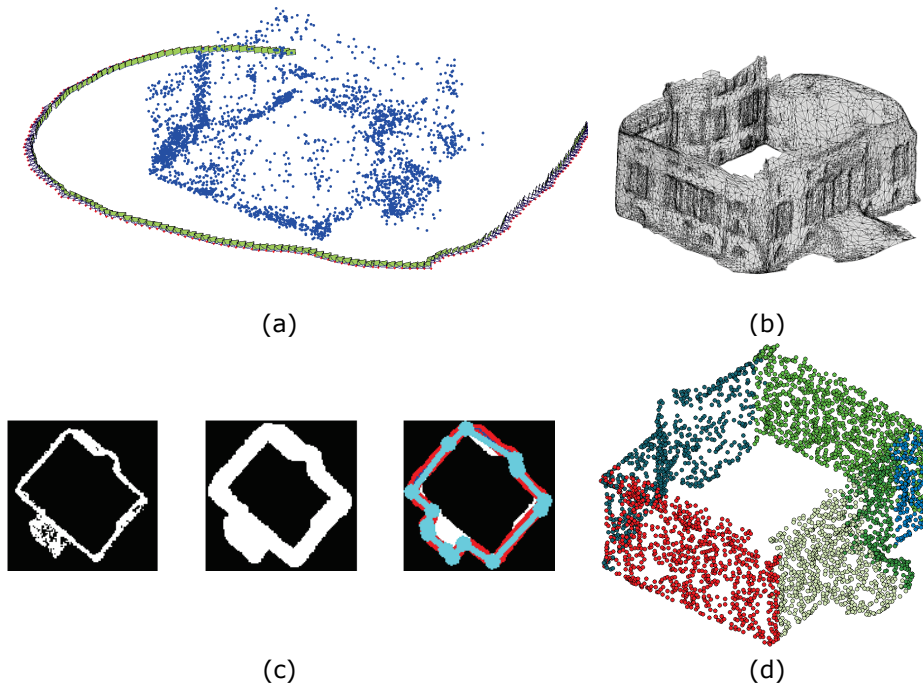


Fig. 4.28: (a) The video imaging track around the building. (b) The derived rough model and mesh. (c) The image morphological operation before filtering. (d) Subdivision result.

The camera network design is implemented per facade and results in a total of 158 cameras within six blocks with the addition of the connecting cameras as shown in Fig. 4.29. This design is automatically applied with a theoretical standard error σ_{xyz} of 10mm in the object space. The distance between the cameras and the object of 10 meters is designed to satisfy the required accuracy (Wenzel et al. 2013). This is based on assuming an image coordinates measurement precision of 10 μm .

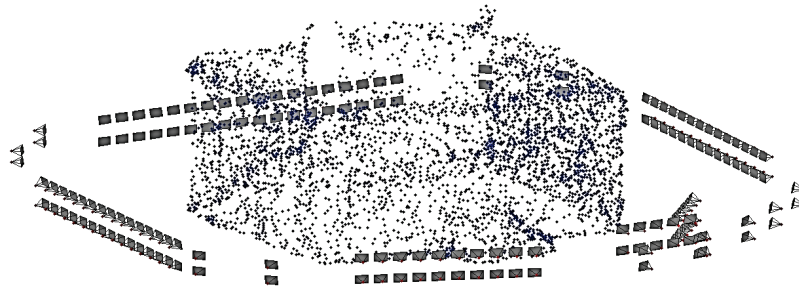


Fig. 4.29: The building initial dense imaging block before filtration.

Hence, the dense camera network is filtered according to both methods of the coverage requirements of minimum three cameras and the filtering for accuracy, as discussed in section 4.5, and results with the imaging configuration of Fig. 4.30a, and 4.30b, which consists of 81 and 86 cameras respectively, after the addition of the connecting cameras. The optimization algorithm is optionally followed after filtering for accuracy (Fig. 4.30b) to find the final optimal imaging configuration as shown in Fig. 4.30c. The optimization implemented on each sub-network of a facade separately and then gathered to compose the total network (Fig. 4.30c). The connecting cameras are then added to complete the camera network where two cameras are added between each adjacent detected facade.

It can be noticed from Fig.4.30 that a significant reduction in the number of the cameras is found through the two filtering techniques. Moreover, Fig.4.30b shows a better camera distribution offered by the filtering for accuracy technique compared to the filtering for coverage technique in Fig.4.30a.

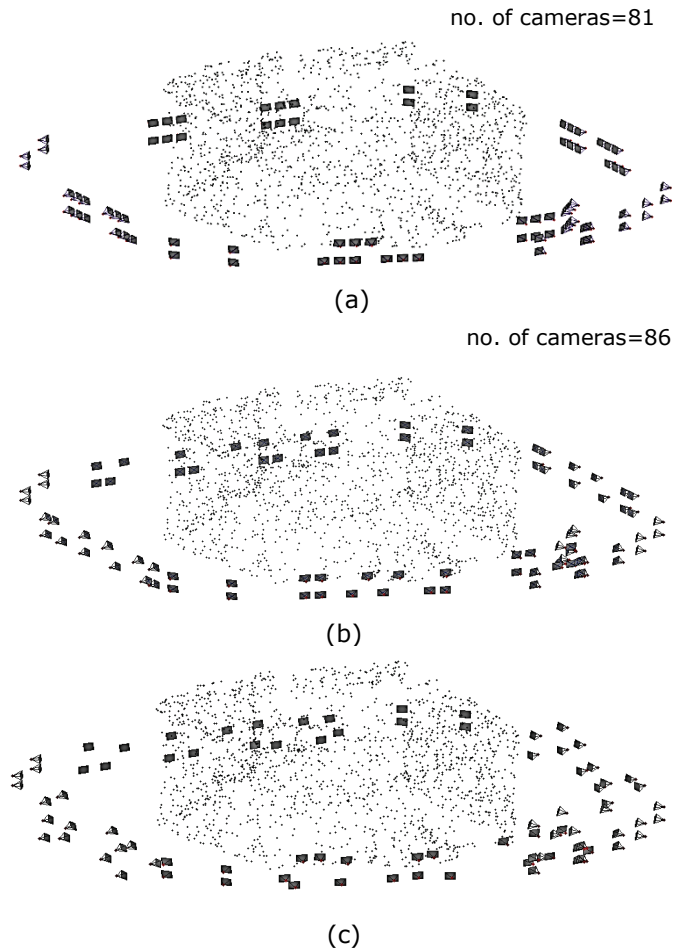


Fig. 4.30: Imaging network configuration of the building. (a) Filtered network for coverage. (b) Filtered network for accuracy. (c) Optimal network.

To validate the network accuracy, error estimation is computed for each point in the cloud and plotted as an exaggerated ellipsoid of errors in the same scale as illustrated in Fig. 4.31. The sizes of the ellipsoids of errors give a good visual indication about the possible errors in the different parts of the object. It can be noticed that the accuracy is degraded after the filtering of redundant cameras (Fig. 4.31b) as explained in section 4.5. However, the optimal network and when the cameras are re-oriented according to constraints and the accuracy slightly improved as shown in Fig. 4.31d. This accuracy improvement is achieved despite having the same number of cameras and the frequency of coverage (Fig. 4.33) of the filtered network. It is worth to mention that the maximum standard deviation indicated in Fig. 4.31 refer to the largest axis of the ellipsoid of the error of the points. The average point precision is also listed in Table 4.1.

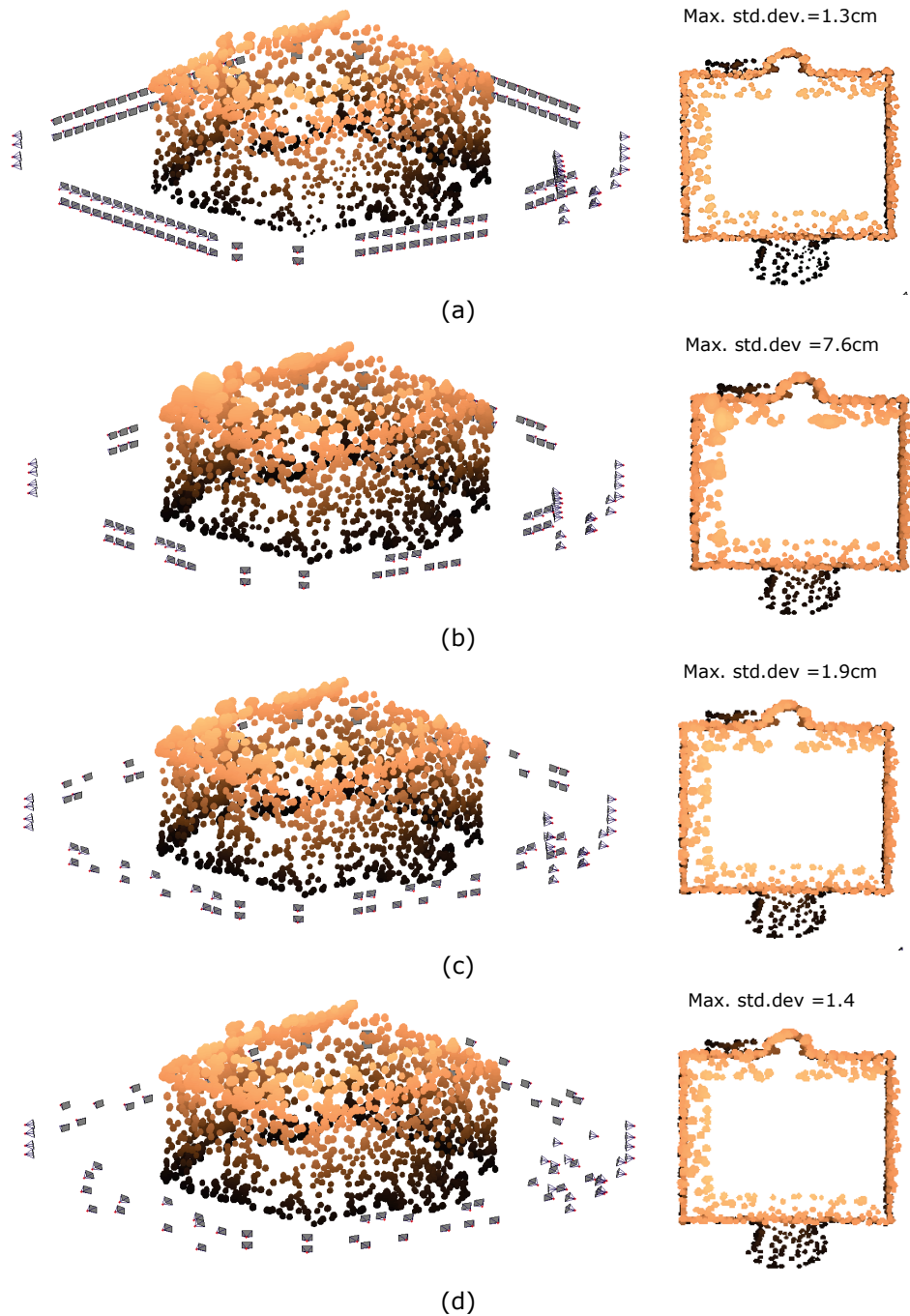


Fig. 4.31: The exaggerated error ellipsoid plot for the building experiment. (a) Dense network. (b) Filtered network for coverage. (c) Filtered network for accuracy. (d) Optimal network.

The optimization run is shown in Fig. 4.32 for each detected façade and a maximum function evaluation count of 1000 is selected for the stopping criteria. Otherwise, the solution is stopping with less iterations if the local minimum and the constraints are satisfied.

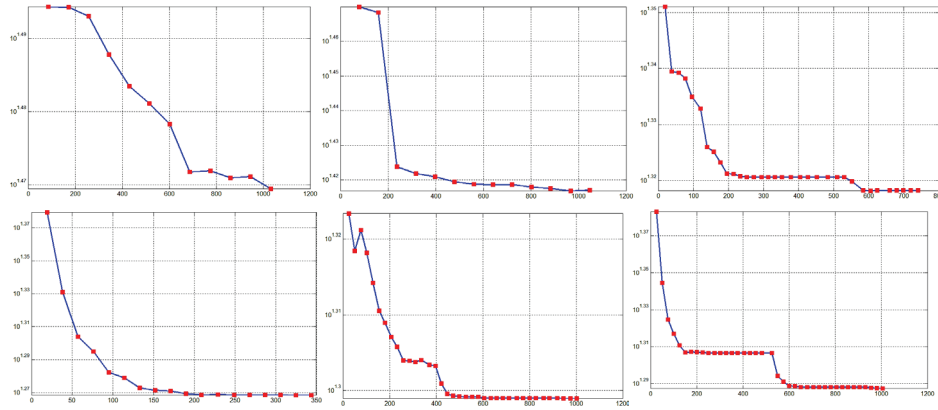


Fig. 4.32: Log plot of optimization runs for each detected façade of the building.

Fig. 4.33 illustrates the amount of coverage for every point in the dense ,filtered and optimal networks respectively. The low number of covering cameras is located in the upper parts of the building facades where the visibility is limited.

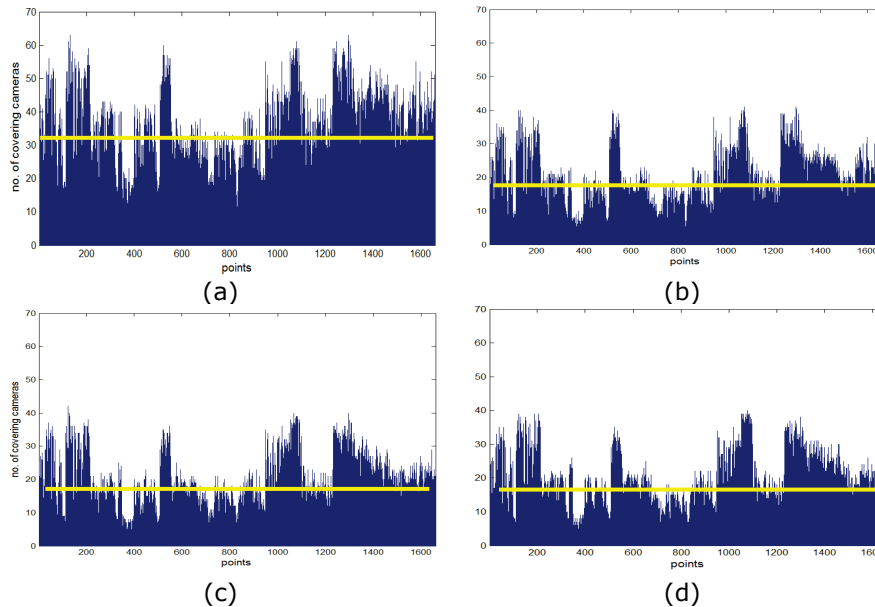


Fig. 4.33: The number of the viewing cameras per point with the average in a yellow line. (a) The coverage in the dense network. (b) The coverage of the filtered network for coverage. (c) The coverage of the filtered network for accuracy. (d) The coverage of the optimal network.

4.9.3 Statue

The second test is implemented on a statue point cloud (283 points) which is generated automatically by the SfM technique from the video imaging stream of the statue by using Boujou software (Vicon 2010) as illustrated in Fig. 4.34a. The point cloud is turned into a surface grid triangular mesh by using Meshlab (2010) to test the visibility with the face normal (Fig. 4.34a). A simulation of a dense block of 62 cameras in two strips with a B/D ratio of 0.2 is designed around this statue in a circular domain. The accessible height of the camera operator is assumed as mentioned previously to be in the range of 0.5-2 meters with a depth-distance of two meters to satisfy the required nominal accuracy of 2 mm. The designed dense camera network is illustrated in Fig. 4.34b where the cameras are oriented in the portrait pose which is more suitable to cover this kind of objects extending in the up (z) direction.

Consequently, as in the previous test, the algorithm of filtering either for coverage or accuracy starts to find the minimal camera network by cancelling one redundant camera iteratively. In this experiment, the filtering for coverage is used which is stopped after the filtering of 28 cameras, and results in 35 necessary cameras. The remaining cameras guarantee the minimum of three intersecting rays for each point with an average B/D ratio of 0.3. The optimization algorithm is followed to find the final optimal imaging configuration, which satisfies all the aforementioned constraints of section 4.7.2 as shown in Fig. 4.35b. The bounds of the camera orientation parameters are selected to be in the range of one meter of the designed camera coordinates and five degrees of the initial angular orientation. This is a matter of user input basing on the object complexity, processing time and accessible space around the statue.

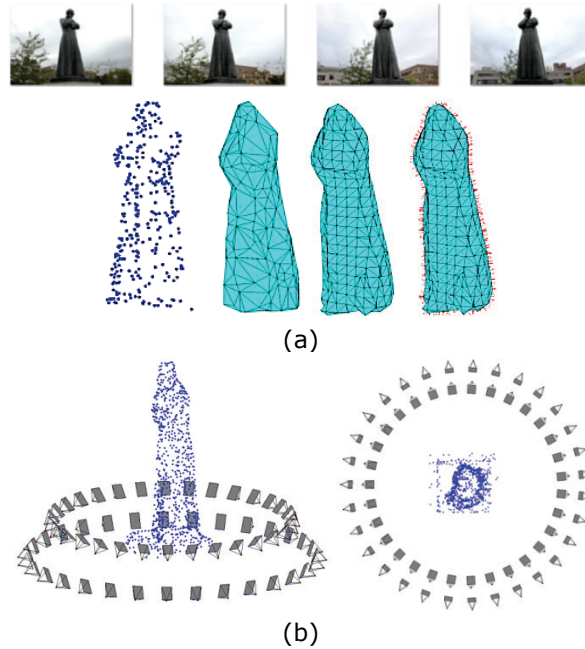


Fig. 4.34: (a) Statue surface mesh. (b) The dense imaging block.

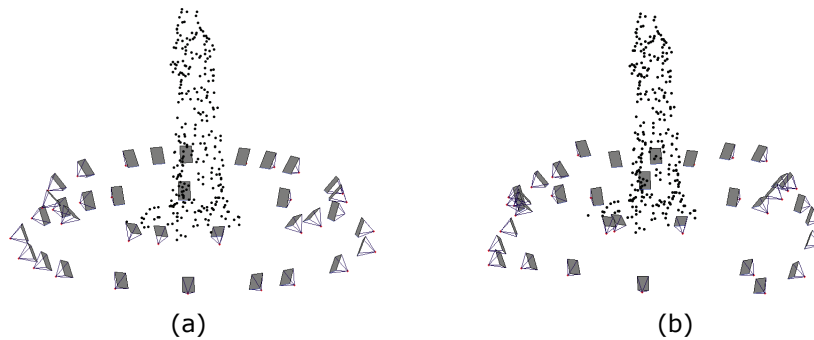


Fig. 4.35: Statue imaging network. (a) Minimal network before optimization. (b) After optimization.

To validate the network strength, internal error assessment is computed for each point in the cloud of dense, filtered and the optimal network. The exaggerated ellipsoids of errors are illustrated in Fig. 4.36, and reported in Table 4.1. The precision of the image coordinates is simulated with a $10 \mu\text{m}$ as mentioned earlier.

It can be noticed in a similar way to the building experiment that the errors in the upper part of the statue are larger than the errors in the lower parts. This is a result of the weak ray intersection and the limited visibility from a layman height.

It is worth to mention the time spent for processing in the statue case. The filtering with respect to the coverage requirements method consumed 25 seconds to reach the minimal network of 35 cameras. The optimization runs for 9 iterations before meeting the stopping criteria as shown in Fig. 4.37, and consumed 14 minutes for processing on Dell latitude E6540 with Intel processor Core i7 of 2.7 GHz CPU. However, different factors like the programming language and the proficiency to prepare the code influenced on the time needed for the implementation. It must be noted that the stopping criteria can be determined either by setting the number of iterations or by setting the threshold of the minimum step length which is usually embedded in the optimization algorithm. Finally, the coverage amount of points before and after the filtering and optimization is shown in Fig. 4.38 and listed in table 4.1.

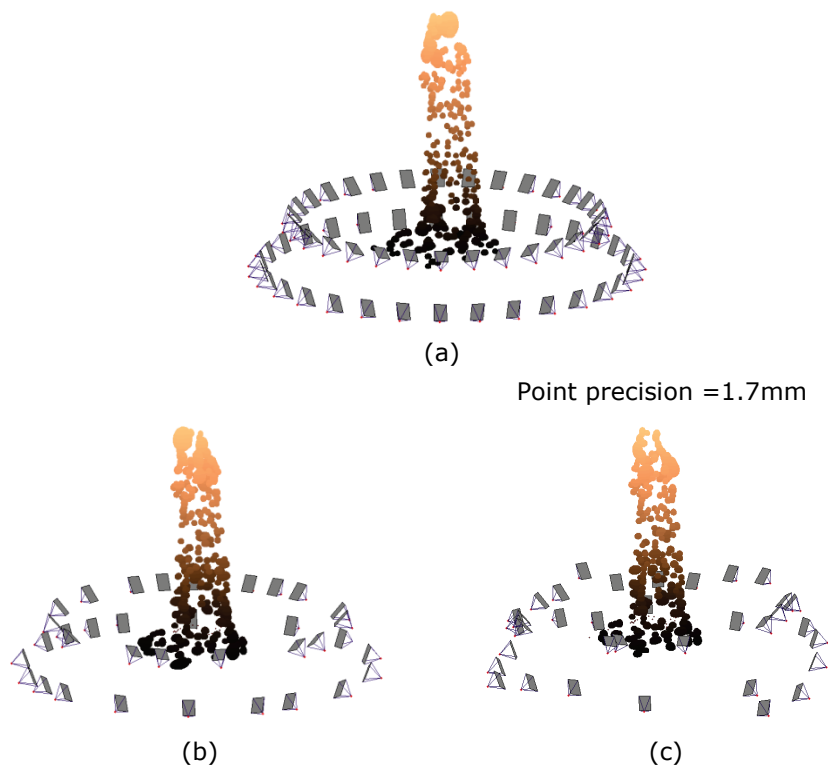


Fig. 4.36: The error ellipsoids for the statue points in three cases. (a) Dense network. (b) Filtered network. (c) Optimal network.

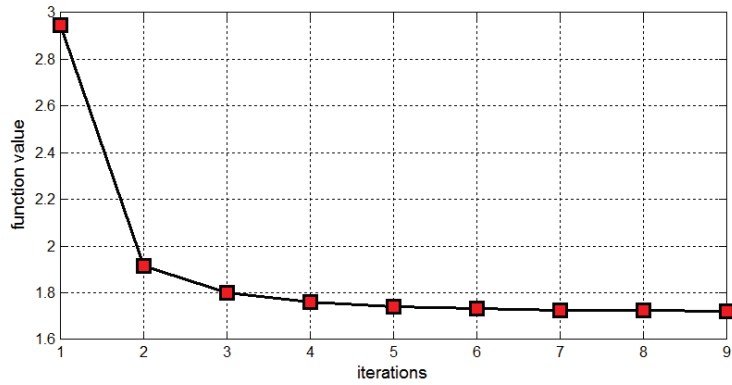


Fig. 4.37: Log plot of optimization runs of the statue imaging.

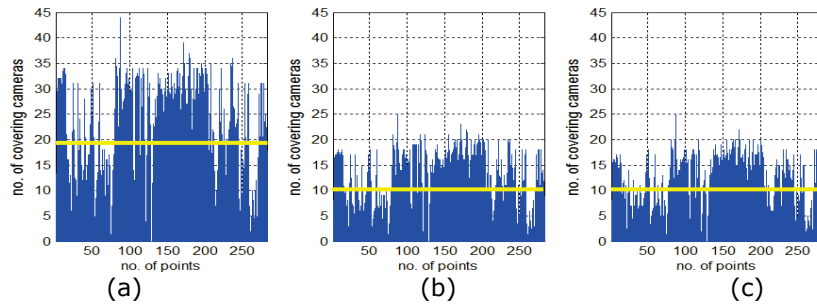


Fig. 4.38: The number of viewing cameras for each object point. (a) Dense network. (b) Filtered network. (c) Optimized network.

It is worth to mention that points with weak coverage (<5 cameras) are located near the statue head where either the visibility is limited or a result of the poor ray intersection geometry.

4.9.4 Discussion of results

The summarized results of the two cases of a building and a statue in terms of estimated precision, mean coverage and the number of cameras are shown in Table 4.1.

Table 4.1: Summary of results.

Type	Cameras	Point precision [mm]	Average coverage (cameras)
Building case study			
Dense block	158	1.7	31
Filtered network (Coverage)	81	2.5	17
Filtered network (Accuracy)	86	2.4	17
Optimal network	86	2.2	17
Statue case study			
Dense block	62	1.0	19
Filtered network (Coverage)	35	2.9	10
Optimal network	35	1.7	11

The summarized results of both experiments showed several benefits of the proposed camera network design.

Table 4.1 shows a high percentage of average coverage for the building and statue cases after filtering. This coverage will guarantee a minimum of three cameras per point which is suitable for the task of 3D modelling. The reduced number of cameras after filtering will offer a significant reduction in the processing time for 3D modelling and SfM while keeping the sufficient coverage of points. We presented in our research paper (Alsadik et al 2014a) a comparison between dense and reduced networks in terms of time consumption which showed a 60% time reduction.

For the building test, the estimated maximum error in the point cloud showed that filtering for accuracy is better than using filtering for coverage. However, the time needed for filtering to accuracy method is twice the time for filtering to coverage method. This is logical because of the need to compute the covariance matrix for every point when using the method of filtering for accuracy.

Optimization also introduced a slight improvement in the accuracy obtained in the filtered networks in both experiments. The computations in the two experiments showed a stable convergence of the solution to optimal minimum. However, optimization is a time consuming procedure and the experiments showed that the camera networks after filtering mostly satisfy the required accuracy. Therefore, optimization is better to be considered when the error after filtering is exceeding the acceptable error limits of a project.

4.10 Summary

In this chapter, we developed a method to find the optimal camera network configuration for the 3D modelling. We started with a simple iterative camera filtering procedure where a dense, short base camera network is initially designed according to the imaging standards. The dense network is then filtered to reach the minimum number of at least three cameras for each point. Then we used the nonlinear constrained optimization techniques to improve the positional accuracy of the point cloud by adjusting the camera initial orientation to improve the estimated positional accuracy.

Three case studies of a simulated object, a building, and a statue were tested and indicated a significant reduction in the number of the designed initial dense camera networks. The cameras in the first test of a building are reduced after filtering from 158 to 81 and 86 respectively, while for the statue case, the 62 cameras are reduced to 35. This strategy is beneficial to have a good covered imaging configuration with a minimum number of short baseline cameras. Consequently, it is shown in Table 4.1 that despite the significant reduction of the number of cameras after filtering, the accuracy in the three case studies is only slightly degraded.

It can be seen that our optimization approach improved the degraded accuracy after filtering by using optimization techniques, which gives a slightly improved point cloud accuracy and average coverage as shown in Table 4.1. The estimated errors are complying with the required accuracy for the modelling of the building and the statue respectively.

Chapter 7 is devoted to investigate the proposed methodology, including filtering and optimization on a complete experiment with external validation of a cultural heritage church building.

Chapter 5 Guided Image Capture and Post Processing

This thesis chapter aims to introduce a practical, simple, and efficient way to guide the non-professional camera operator to the optimal camera location which is discussed in chapter 4.

However, as mentioned in chapter 1, the question is how the system guides the camera operator to the correct location for capturing the designed images while maintaining the same designed angular orientation.

The key idea as proposed in this chapter is to create synthetic images of the study object based on the optimal orientation.

A synthetic image is an artificial image as it would be captured by a given camera with the optimal orientation. The motive to create the synthetic images is the suitability of these images to guide even non-professionals to the designed image orientation. Therefore, the camera operator can visually interpret the images and try to capture them in reality. Moreover, the synthetic images offer a powerful tool to verify whether the image is taken from the right spot and pointing in the desired direction. Therefore, matching between the real captured images and the synthetic images is applied to decide the amount of equivalence or similarity.

After the completion of image capture, a post processing will be needed to complete the image orientation and the dense matching to have a 3D point cloud. As mentioned in the research problems of chapter 1, image orientation holds many difficulties in the matching phase and the bundle adjustment like in the case of wide baseline configuration, illumination changes, or when repetitive pattern and homogenous texture areas exist.

The suggested improvement in the mentioned post processing, is to:

- 1- Support the SIFT matching by restricting the matching area.
- 2- Guide the image matching to avoid the full pairwise matching method. This means, based on the designed network, to only match images that should be overlapping. This will save time and avoid mismatching with repetitive textures.
- 3- Support the bundle adjustment with the proper initial parameters.

This chapter will be divided into two main parts: the guiding of the image capture and post processing.

5.1 Guiding the image capture

The proposed concept is based on the previously taken video streaming and SfM technique (chapter 4). Although this was used to create, a rough point cloud for the purpose of optimal camera placement, it is also useful to support with the necessary and sufficient image texture of the target object. The key idea is to create multiple synthetic images of the object as seen from the optimal camera locations and viewing directions. In addition, image matching is needed to decide about the amount of equivalence or similarity between the real captured and the synthetic images, i.e. to estimate how well the optimal situation is reached per shot.

Consequently, if the matching is sufficient within the allowed limits as will be shown in section 5.1.1, then the orientation of the real captured images will be computed.

The impact of adding the real image on the total orientation accuracy is validated before proceeding for the next image capture until completing the whole imaging network. Fig.5.1 illustrates the general workflow of the guiding system.

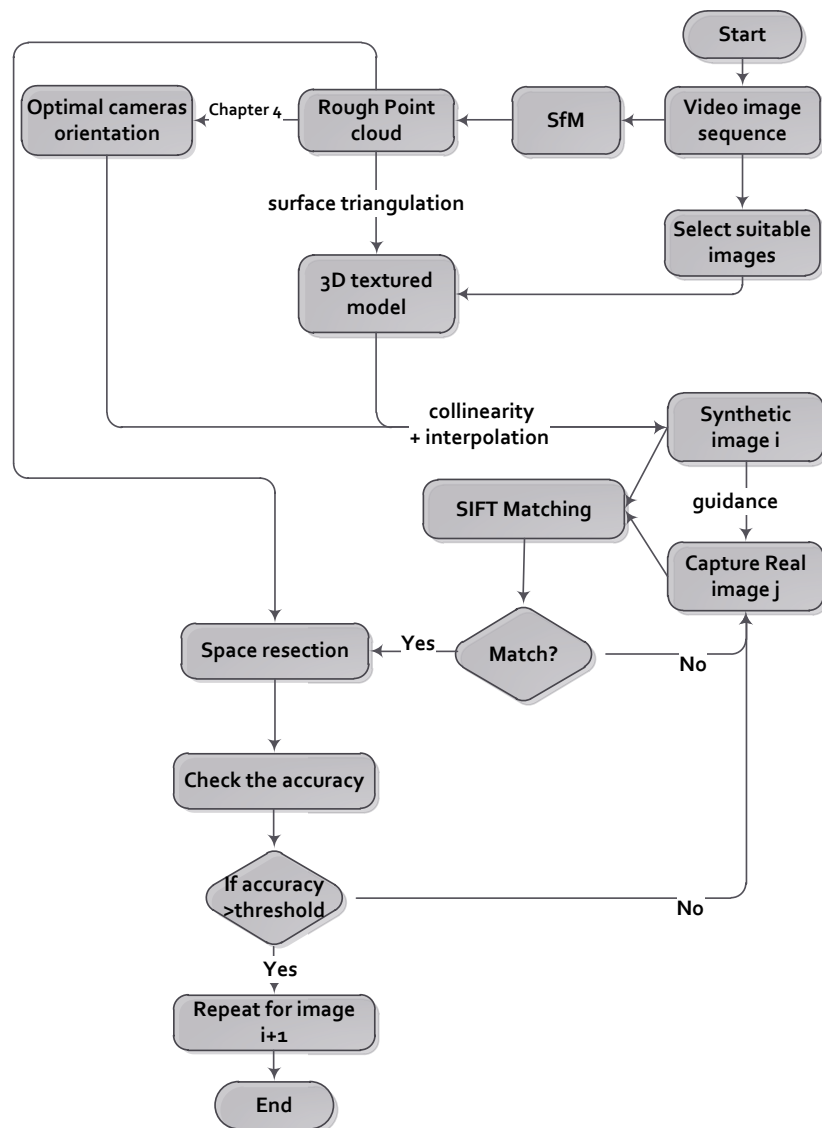


Fig. 5.1: Workflow of the guiding imaging system.

5.1.1 Creating the synthetic images

The first step of creating synthetic images is to compute a 3D triangulated surface of the previously derived rough cloud by an efficient surfacing technique like ball pivoting (Bernardini et al. 1999) or Poisson reconstruction (Kazhdan et al. 2006). This is followed with texturing the mesh by finding the suitable covering pre-oriented video images that were used in the previous step described in chapter 4. Accordingly, an image resampling is implemented to get a textured 3D model as shown in Fig. 5.2. This textured

3D model will be projected, based on the designed orientation parameters, to the desired optimal images to create the synthetic images.

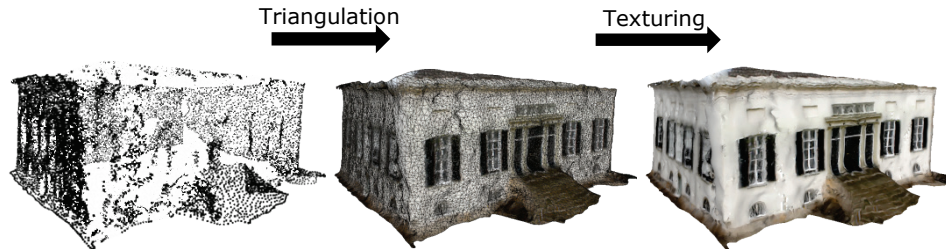


Fig. 5.2: Low-detailed 3D textured model.

The information about the textured 3D model is simply transformed in two steps back to the planned synthetic images. The first step is to project the 3D coordinates back into the 2D pixel coordinates and the second step is to assign the texture for each triangular face. Therefore, each textured triangular face (three vertices and the patch RGB colour) is transformed as illustrated in Fig. 5.3 from the texture image to the corresponding synthetic image by linear interpolation.

The transformation is applied to each face by moving across a bounding rectangle and assigning the pixel value from the texture image to each pixel that falls inside the triangle. It must be mentioned that a low resolution (e.g. 640*480) pixel is chosen to compose the synthetic images. This low image resolution consumes less processing time to comply with the fast implementation needs.

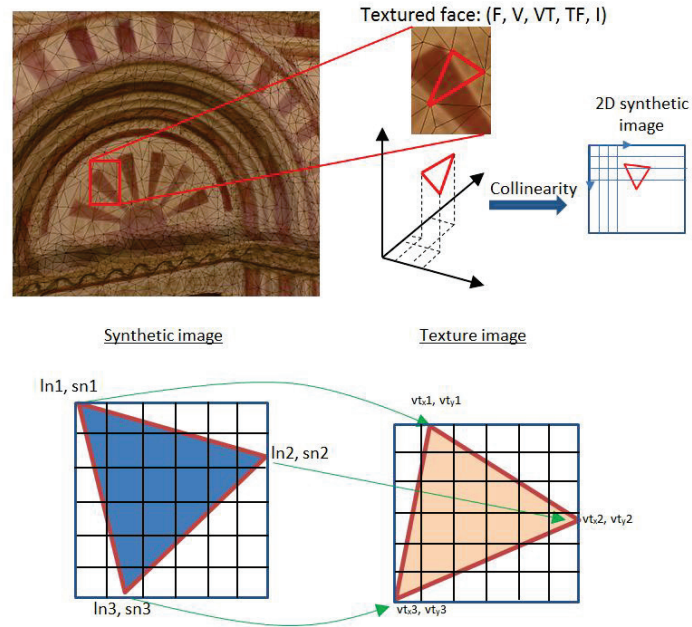


Fig. 5.3: Texture transfer of synthetic images.

An additional visibility condition is added to avoid self-occlusions. This is applied by using the z-buffering technique (chapter 2). Fig. 5.4 and Fig. 5.5 illustrate the method of creating free self-occluded synthetic images.

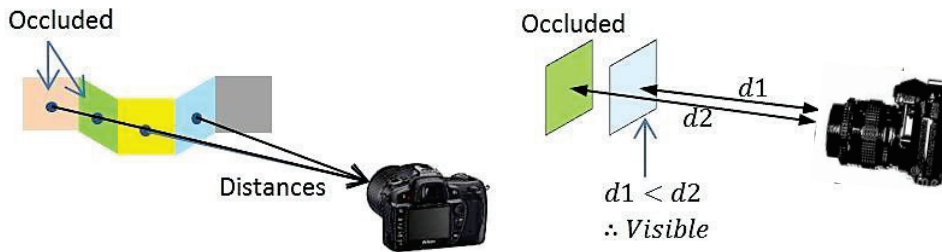


Fig.5.4: Removing occluded pixel values by distance buffering.

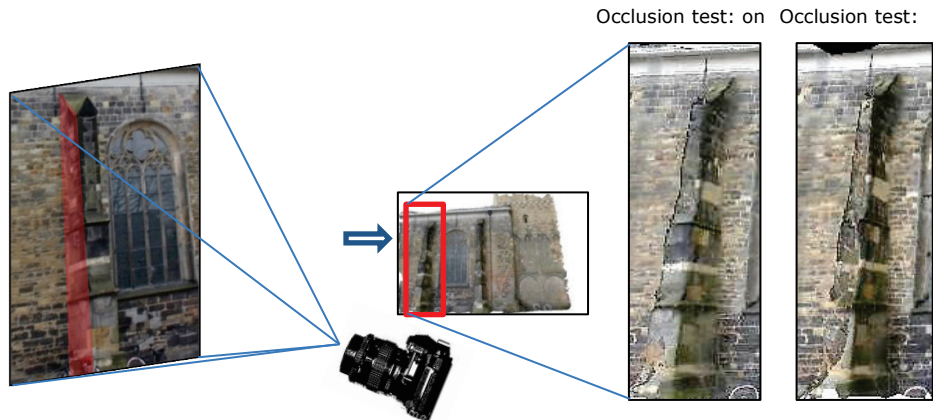


Fig. 5.5: Synthetic image creation with and without self-occlusion computation.

5.1.2 Real image capture

After the creation of synthetic images, as shown in the previous section, the images should be captured in reality. The captured HR images should be matched in real time with the synthetic images to decide if they are taken correctly within a threshold limit. Further, every image is re-captured if it doesn't match accurately until capturing the proper one.

In practice, the guiding of image capture can be designed by connecting the camera to a laptop. Then, to use software that is capable to show the live view synchronized images like Eos Utility software for Canon cameras as shown in Fig. 5.6. This will enable the camera operator to match visually the camera live view with the synthetic image for real image capturing. Subsequently, every captured image is stored in a folder and also tested if it matches the synthetic image and preserves the accuracy as will be discussed in the following sections.



Fig. 5.6: Software-hardware illustration of the guiding system.

5.1.3 Matching between real and synthetic images

To decide whether the captured images are correctly taken and similar to the synthetic images, image matching is applied by using the scale invariant keypoint detector of SIFT (Lowe 2004). The reduction of the image resolution to the same synthetic image size is beneficial to reduce the processing time of the image matching to satisfy, as mentioned, with the fast implementation needs. However, the matching probably suffers from some blunders caused by the poor texture of some object parts in the synthetic images, which can deteriorate the results. These blunders can be filtered according to the following strategy:

- **Restrict the matching space with respect to the synthetic image foreground**

Synthetic images based on the derived model as mentioned in section 5.1.1 will have a white background. Since the images should be taken, nearly, in the same pose, the matching space is restricted. The restriction is based on the convex hull of the detected keypoints in the synthetic image. This convex hull will define approximately the outer boundaries of the study object texture as illustrated in Fig. 5.7. Then, all the keypoints of the real image that are detected outside the convex hull will be neglected for matching. This is beneficial to reduce the number of wrong point matches and to reduce the matching cost. However, some parts of the white background can also be included within the convex hull based on the object shape (Fig.5.7).

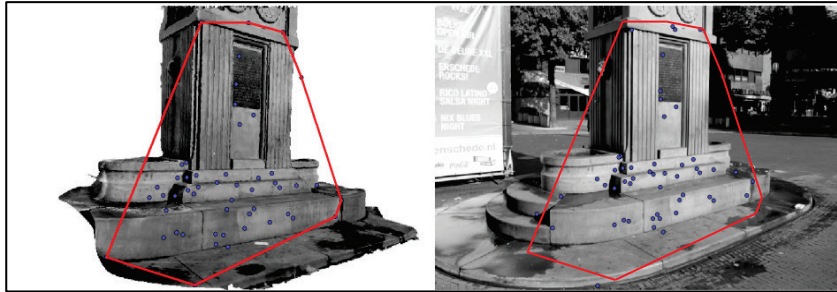


Fig.5.7: Restriction of the matching space. Left: SIFT points in synthetic image. Right: Restricted SIFT points in real image.

- **Match the images by SIFT**

After the initial filtering of background points in the real image, the SIFT matching technique is applied, which is invariant to scale, rotation and illumination change. However, there is still a possibility of mismatching as shown by the yellow lines in Fig. 5.8.

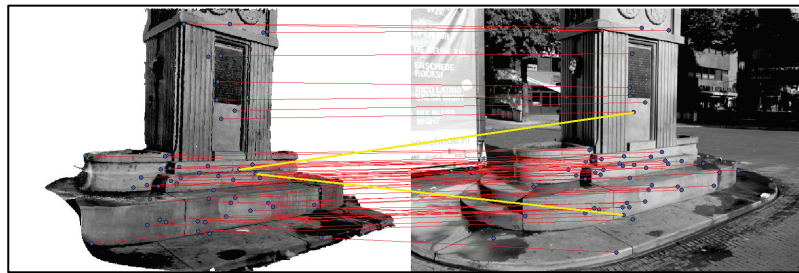


Fig.5.8: SIFT matching between real and synthetic images.

- **Statistical blunders detection and prior removal**

To remove the blunders of the SIFT matching two simple tests are applied as follows:

- Since the images are assumed to be taken, nearly, in the same pose, the corresponding matching lines are expected to be parallel as shown by the red lines in Fig. 5.8. Therefore, the y -parallax between the corresponding points is computed. Then, matches that have exceeded a threshold of 50 pixels of y -parallax will be excluded, which is one tenth of the used image height.
- To support the blunder removal, an additional statistical approach is used. This is simply applied by computing the average distance between the homologue matched points, and filtering out the match distance if it exceeds three times the standard deviation 3σ of the distance. This is motivated by the assumption of having approximately constant distance between equivalent image points of the same orientation. Fig.5.9 illustrates the refined matching result.

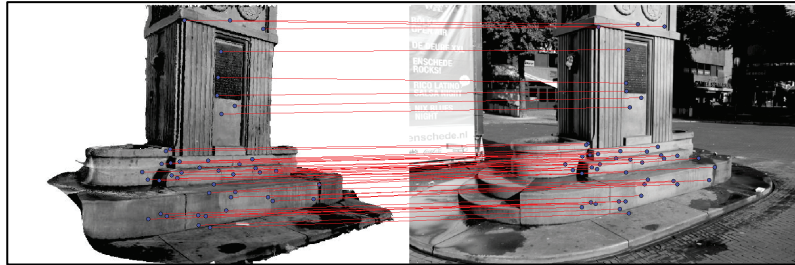


Fig.5.9: The refined points matching between synthetic and real image.

In addition, to validate if the real captured images are accepted, we have to compute the camera orientation by space resection as will be discussed in the next section. It must be noted as shown in Fig.5.9 that few blundered matches still exist, however, the blunder detection by using data snooping (Ghilani and Wolf 2006) will be used in the adjustment problem to ensure an accurate determination of the real-captured image orientation.

5.1.4 The validation of the real captured images

In order to validate the real-captured image before proceeding to the next image, the camera orientation parameters will be computed and compared to the optimal ones. The orientation parameters for a single image are usually computed in photogrammetry by space resection with collinearity equations. The minimum number to apply the space resection computations is three points to solve the image exterior orientation parameters.

However, there is a need to determine the space XYZ coordinates of the matching points to implement the resection. As mentioned in section 5.1.1, every pixel in the synthetic images will have its 3D location as well. However, in case to avoid consuming a large computer memory, another approach can be followed by interpolating, only, the location of the SIFT points:

The XYZ coordinates of the matching SIFT points are interpolated by projecting the 3D mesh to the synthetic images (Fig.5.10a). This is followed by using the inverse distance weight (IDW) interpolation of Equation 5.1. The interpolation is constrained to the triangle that contains the projected SIFT points in the surface mesh to get an accurate result as shown in Fig.5.10b. However, it may happen that the same matching point represents a triangle mesh vertex and then there is no need for the interpolation.

$$X = \frac{\sum_{i=1}^3 \frac{X_i}{d_i}}{\sum_{i=1}^3 \frac{1}{d_i}}, Y = \frac{\sum_{i=1}^3 \frac{Y_i}{d_i}}{\sum_{i=1}^3 \frac{1}{d_i}}, Z = \frac{\sum_{i=1}^3 \frac{Z_i}{d_i}}{\sum_{i=1}^3 \frac{1}{d_i}} \quad (5.1)$$

where d is the distance between the three triangle vertices and the interpolated point.

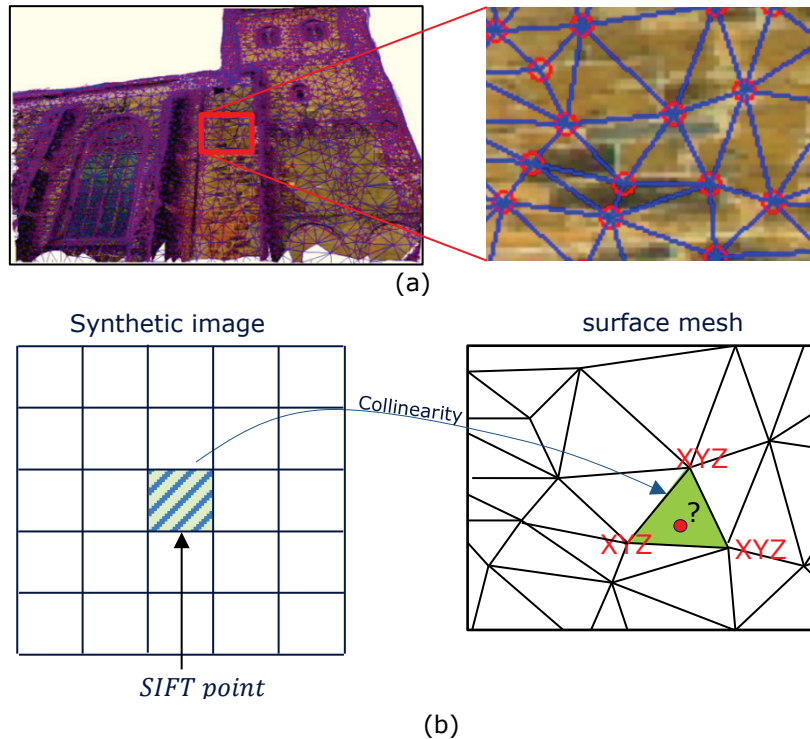


Fig. 5.10: Space coordinates interpolation of the matching points. (a) The surface mesh projected on the synthetic image. (b) The coordinates interpolation.

After computing the XYZ coordinates of matching points, a space resection is applied to the real images with the aid of the data snooping (Ghilani and Wolf 2006). This is due to the possibility of still having a wrong SIFT matching result.

After completing the real-captured image orientation by space resection as shown in Fig.5.11, it is possible to validate whether the image is accepted as the desired optimal image or not. This validation is done by incorporating the orientation of the real image in the imaging network instead of the designed one and then estimating the accuracy of the point cloud by bundle adjustment. Since the accuracy requirement is already defined before starting the project, it is easy to decide either to accept or reject the captured image. Therefore, if the accuracy is significantly degraded, another try for taking the real image is to be done. The process is continued successively until the completion of the whole imaging network.

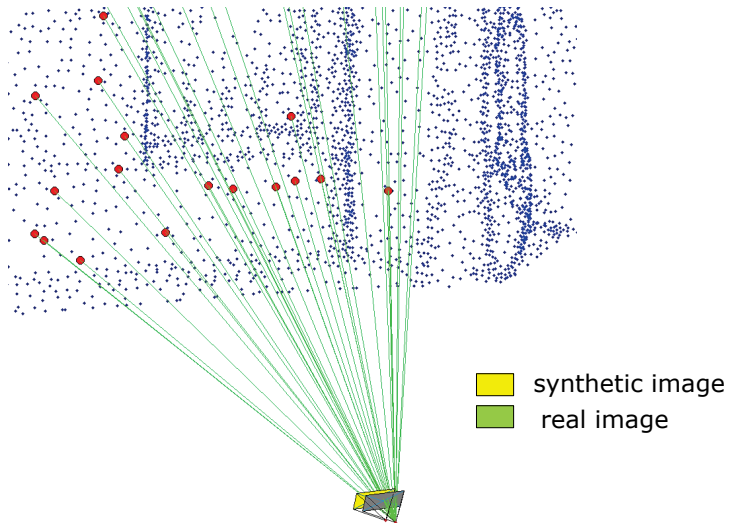


Fig. 5.11: The space resection of the real-captured image

The algorithm of the validation of every captured image is illustrated in Fig. 5.12.

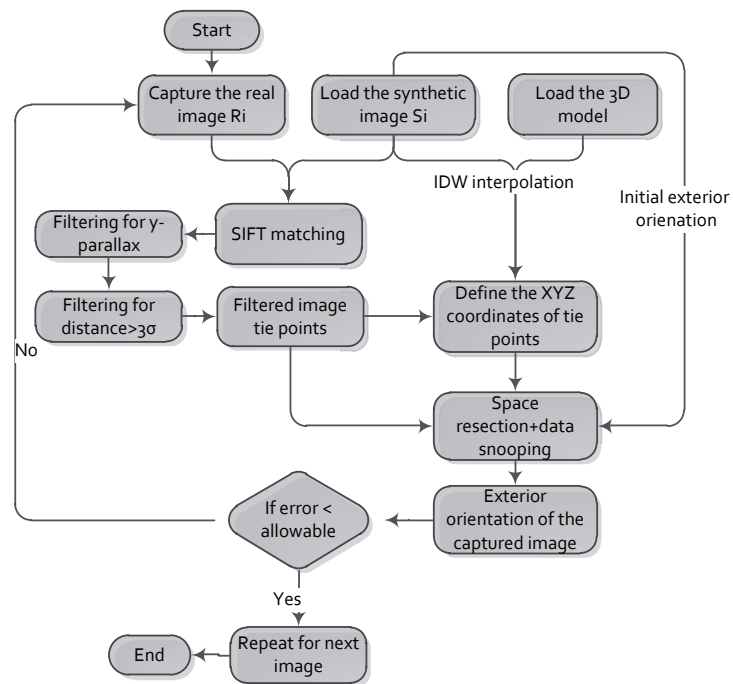


Fig. 5.12: The validation of the real captured image workflow.

Fig. 5.13 illustrates the proposed graphical interface of the guiding system and validation.

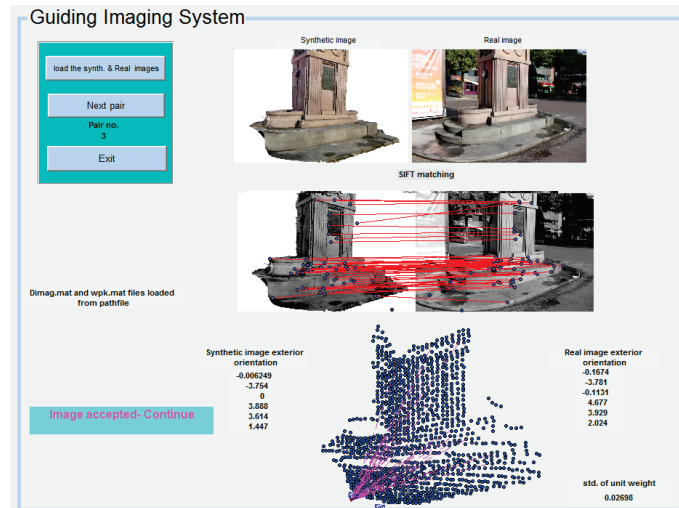


Fig.5.13: The guiding system and validation.

5.2 Post processing computations

After the completion of capturing images as described in section 5.1, the post processing for the camera orientation is followed. First, the overlap connection between the images is to be identified by what is called the connectivity graph (chapter 1). Second, the use of the scene geometry and the initial camera orientations for a robust tie point matching and bundle adjustment will be presented.

5.2.1 Image connectivity for matching

The image connectivity is recomputed at this stage based on the real captured images with their new orientation which is slightly different from the designed orientation. These differences could lead to a change in the image connectivity.

The image connectivity of matching in our approach can be extracted by projecting back the sparse object points that are used in the planning steps. This helps to decide their visibility status within the captured images by using:

- The designed exterior orientation parameters.
- The surface points and their normal directions.
- The interior camera parameters.

Therefore, the decision of having two overlapped images is made by testing the existence of shared points that are visible in both images. Fig.5.14 illustrates the image connectivity graph for the guided matching of 106 images around a building (Alsadik et al. 2013b). The very dense tree of the large set of images indicates the necessary huge amount of data to be handled and processed in the full pairwise matching.

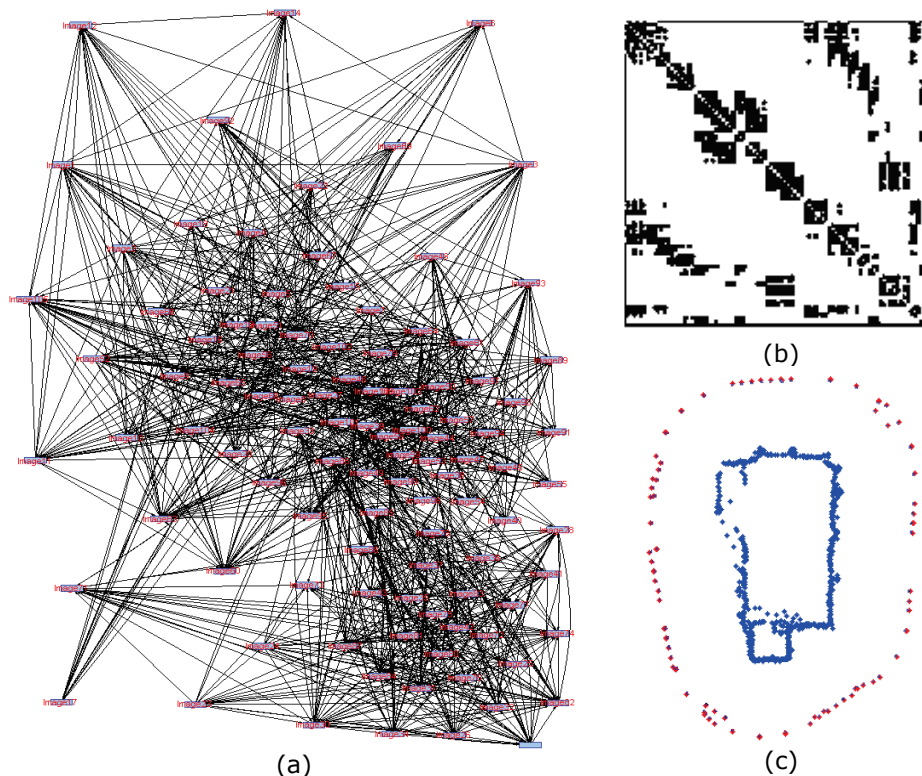


Fig.5.14: (a) The image connectivity graph. (b) The matching matrix. (c) The pre-planned network of 106 images.

The mismatching of pairs in the image connectivity can result in a wrong orientation of the SfM pipeline and bundle adjustment. Therefore, it is not only a problem of the wide baseline imaging, but also the effect of repetitive pattern on the full-pairwise matching results where the symmetric pattern in the object can result in mismatching. Fig. 5.15 shows an example of the wrong orientation of an image sequence when using the SfM technique despite the use of RANSAC for outlier rejection. The repetitive pattern of the fountain sinks in Fig. 5.15a affects the tie point matching which can be avoided by guiding the connectivity between the captured images (Fig. 5.15c). In addition to the successful orientation, guided matching offers a shorter time for processing because of the reduced number of image pairs to be tested for matching.

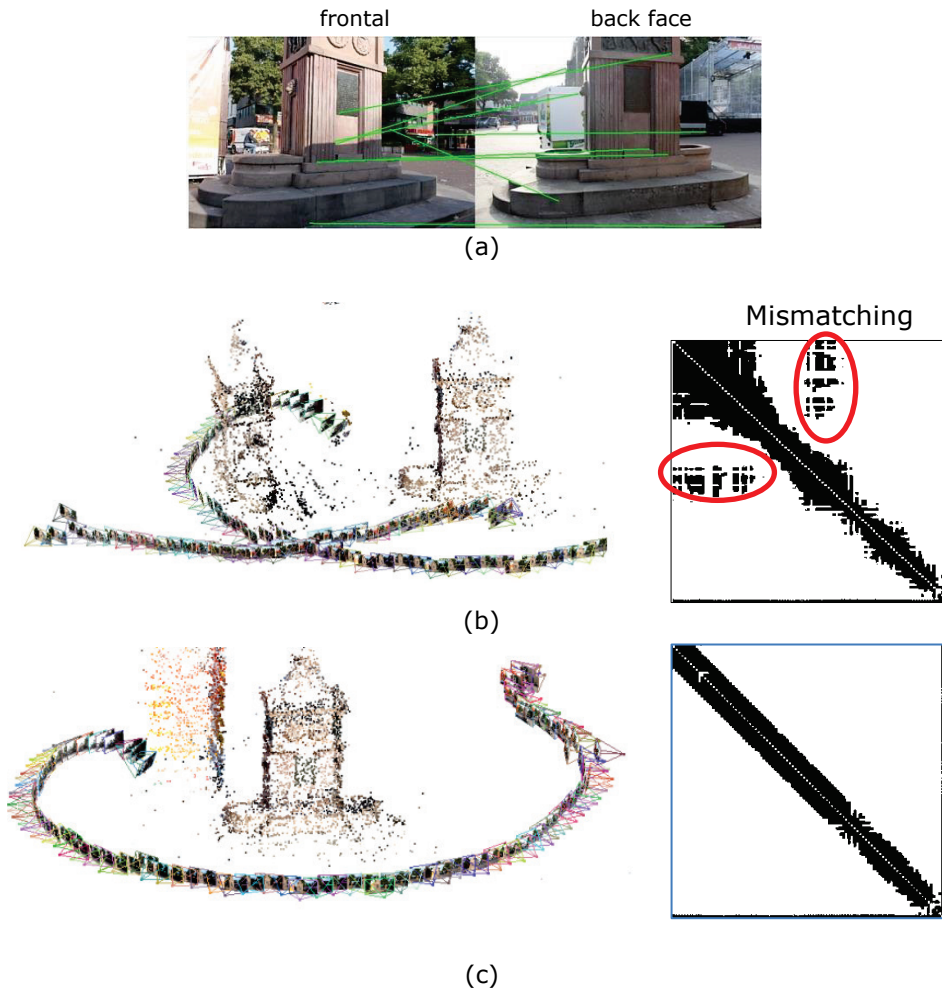


Fig.5.15: Mismatching in the full pairwise approach leads to unsuccessful orientation. (a) The image pair. (b) Full pairwise approach (VSfM). (c) Guided approach (VSfM).

5.2.2 Guided image matching by exploiting the object model

Once the image connectivity is defined, the tie point matching is only applied in the stereo pairs which are linked within the connectivity graph. Moreover, we propose a method to invest the pre-knowledge of camera orientation and scene geometry. The approach is to use the derived rough 3D model as will be shown in this section. This approach should support the matching by providing a very good approximation for the corresponding tie points in the stereo pairs and reduce the processing cost compared to search the full image resolution. In addition, possible problems due to symmetries or repetitive textures occurring at opposite façades will be avoided (e.g. Fig.1.4 and Fig.5.15).

The rough 3D scene geometry is already employed in guiding the camera operator in section 5.1 where each synthetic image is created and also assigned the XYZ scene information per pixel. This 3D information will be transformed to the real images as follows:

For each real image, a 2D affine transformation is applied because it does not fit exactly to its equivalent synthetic image. However, HR real images are first downsampled to the same size of the synthetic images before the transformation. The parameters of the transformation are computed using the detected corresponding keypoints in both, the synthetic and real images. The XYZ information is then passed from the synthetic image (section 5.1.1) to the keypoints of the real image (Fig. 5.16).

The next step is to apply the tie points matching based on the predefined connectivity (section 5.2.1) and the XYZ of the keypoints as well as the approximate exterior orientations. The proposed guided matching is followed by projecting each keypoint location back into the full resolution image by scaling. Then, search windows are defined around the keypoints where the tie points matching will be restricted as shown in Fig. 5.16. The window size is a user-defined parameter and depends on several factors, including the full resolution of the images, the initial image orientation, and the density of the tie points. The experiment in section 5.3 compares SIFT, SURF (Bay et al. 2008) and the normalized cross correlation (NCC) method.

Furthermore, blunders are filtered through a RANSAC procedure using an F-Matrix estimation. Accurate matching will certainly improve the image orientation results and the 3D modelling task at the end.

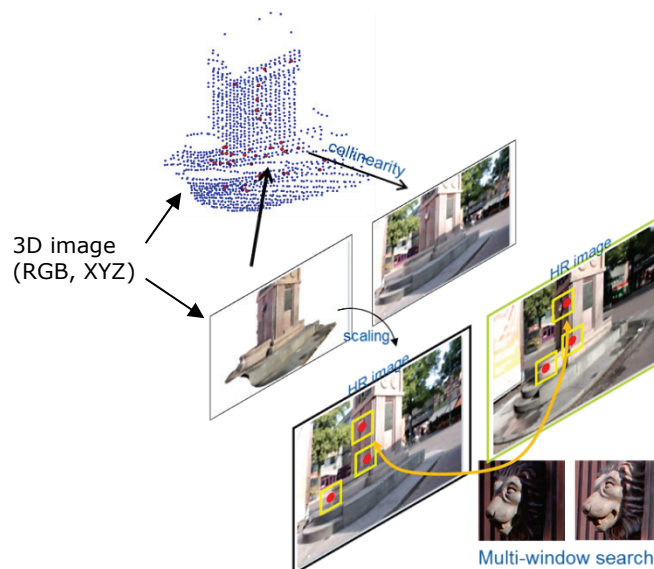


Fig.5.16: Keypoint estimation in high resolution images.

Fig. 5.17 shows the proposed method of robust correspondence matching (Alsadik et al. 2013b).

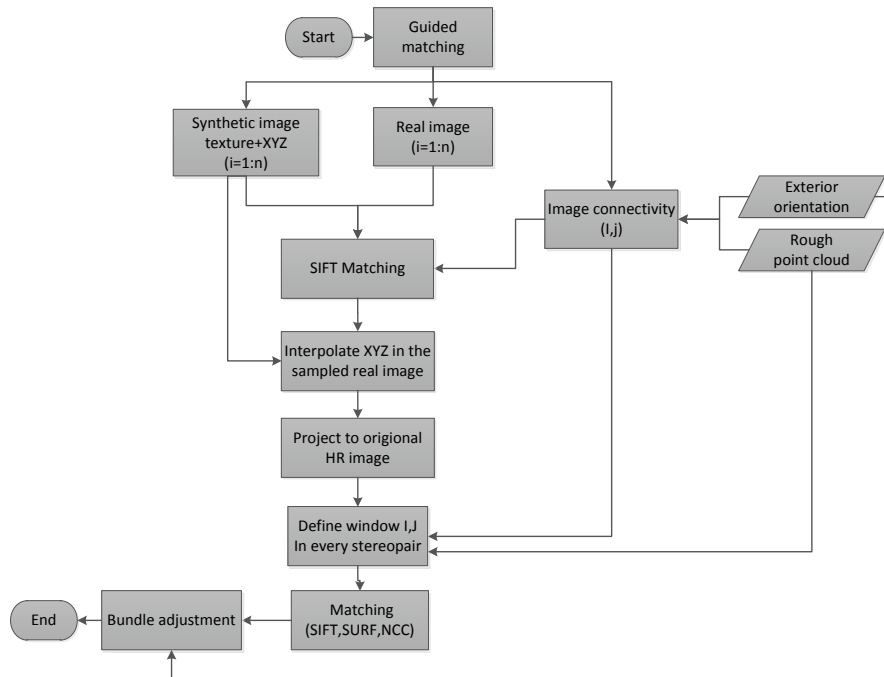


Fig.5.17: The guided matching methodology by exploiting the 3D model.

The methodology can be summarized as follows:

- 1- Create synthetic images according to the designed orientation parameters as described in section 5.1. Every synthetic image is represented by a 2D texture image and 3D information of the pixel coordinates.
- 2- Develop an image connectivity tree based on the pre-knowledge of the object rough model and the exterior orientation of the images.
- 3- Implement a SIFT matching on both the synthetic image and its corresponding real captured image for every stereopair in the image connectivity graph. The two dimensional relation between the SIFT points in both equivalent images will be used to transform the 3D coordinates (XYZ) from the synthetic image to SIFT points to the real image.
- 4- Apply the tie points matching between the real full resolution images in every stereopair. In the left image, every scaled SIFT point will be used to define a small sub-image w_1 . In the right image, the same 3D coordinates of the SIFT points will be projected back by using the pre-designed exterior orientation. The pixel coordinates of the projected coordinates will be used to define the same size of windows w_2 . The SIFT

matching between every pair of windows is named here as the guided SIFT matching.

5.3 Experiment

The proposed method was tested on a monument in the old city of Enschede in the Netherlands. This cultural heritage object was built in 1912 to commemorate the disaster of the city fire in 1863. A low detailed, textured model is derived after processing a video image sequence of the monument (Fig.5.15c) using open source VSfM, SURE and Meshlab. Then, a camera network is designed as a circular domain around the monument point cloud. Accordingly, a two strip dense ordered imaging block of 58 cameras is first designed (chapter 4) as shown in Fig.5.18. Then, the imaging network is filtered according to accuracy demands and the cameras are reduced from 58 to 30 as illustrated in Fig.5.18

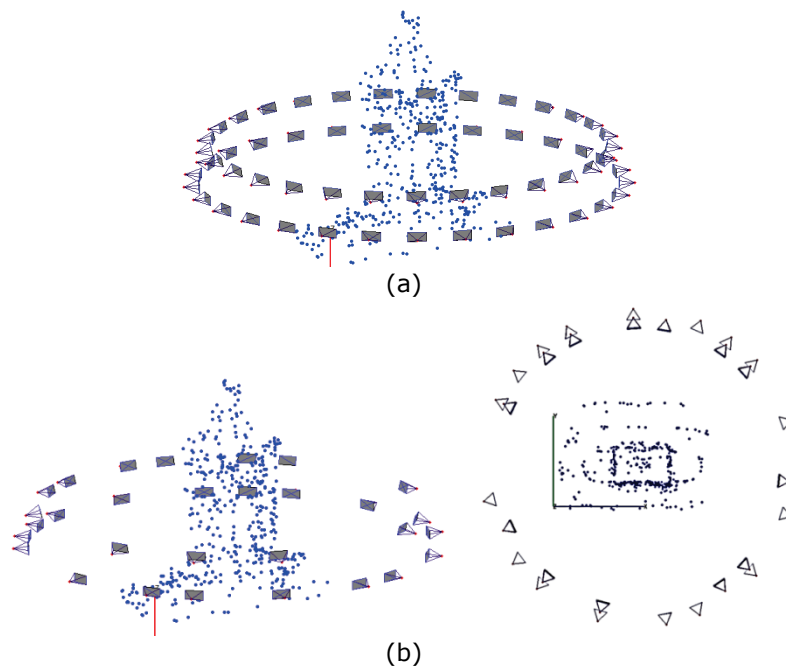


Fig. 5.18: The camera network design of a monument. (a) Dense network. (b) Filtered network from two views.

The error analysis (chapter 4) shows that the filtered network meets the pre-defined accuracy of 10mm. The synthetic images are created in the same way as described previously in section 5.1 as shown in Fig.5.19a. The real-captured images are taken as described in section 5.1.4 and the results are shown in Fig.5.19b. For verification and sharing, the images are also

uploaded into the Photosynth software (Photosynth 2012) and the link of Fig.5.20 illustrates the automated successful image orientation.

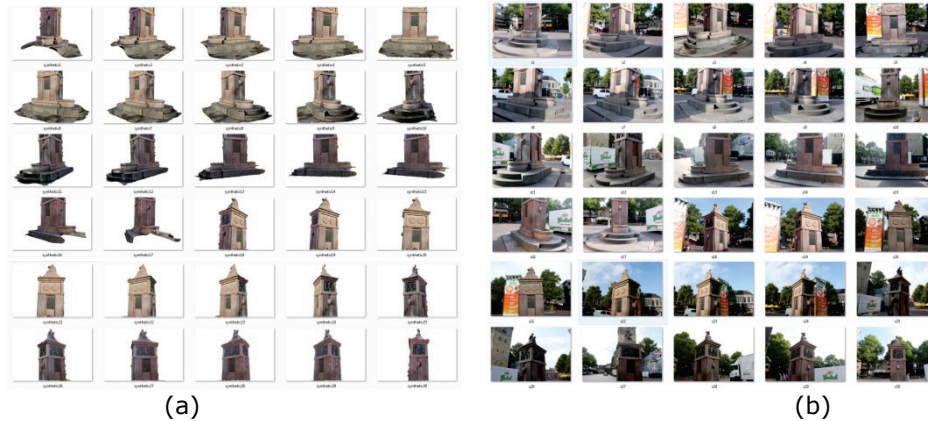


Fig. 5.19: (a) The synthetic images of the monument. (b) The real captured images.

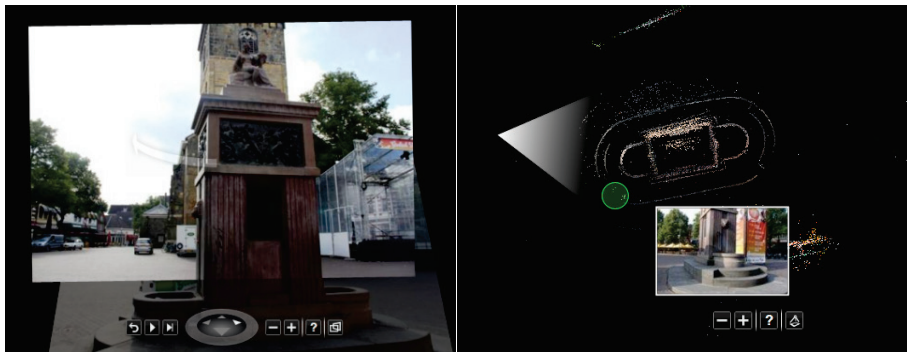


Fig.5.20: The oriented monument images shown in the web service of Photosynth. <http://photosynth.net/edit.aspx?cid=926d20e6-7712-4596-b251-4a306be0d346&wa=wsignin1.0>

To investigate the advantage of our guided matching in the sense of time, the images are resized into four different sizes. The VSfM software is used to implement GPU SIFT matching of the 30 monument images in both cases of the full pairwise and the guided matching. The results are shown in the chart in Fig.5.21 and Table 5.1 respectively. Obviously, the time needed to process the images for the guided matching is about 50% less than full pairwise matching using the same computer. This reduction in processing time is proportional to the number of images as will be shown in the case study of chapter 7.

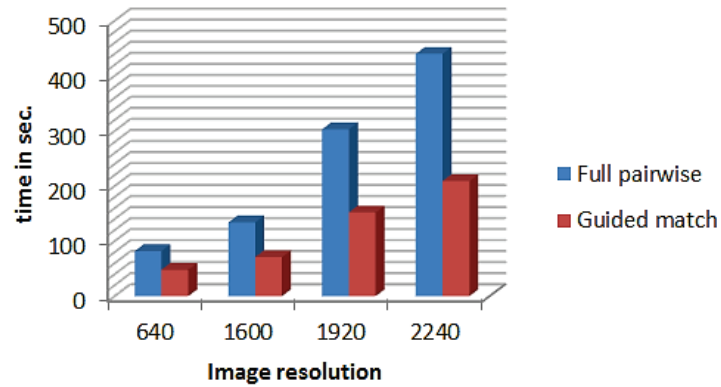


Fig. 5.21. The histogram chart of the processing time for the full pairwise and the guided matching.

Table 5.1: The processing time for the full pairwise and the guided matching.

Resolution [pixels]	Processing time [seconds]	
	Full pairwise	Guided match
640	82	48
1600	134	71
1920	303	152
2240	440	209

Moreover, to analyse the efficiency of the proposed method, four overlapped images are selected (Fig. 5.22) using three different techniques for matching: SIFT, SURF and NCC. The images are selected to have both wide baseline and short baseline configuration. The oriented cameras and their effect on the errors in the object space is shown in Fig. 5.23.

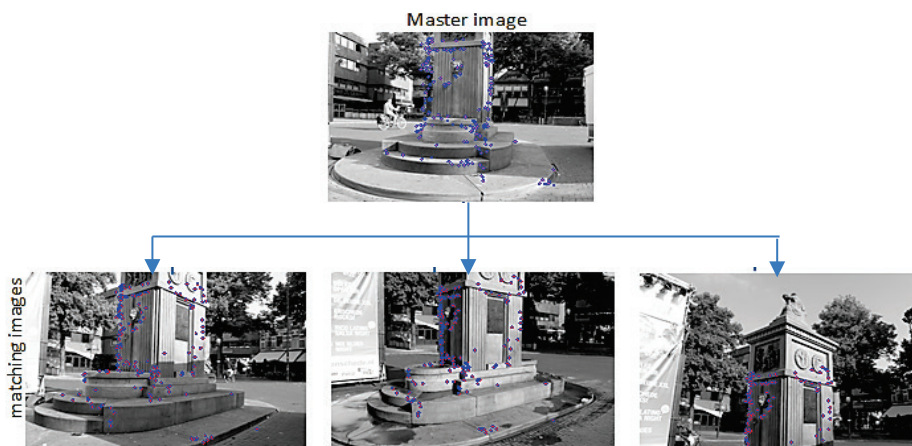


Fig.5.22: Four-image sample set.

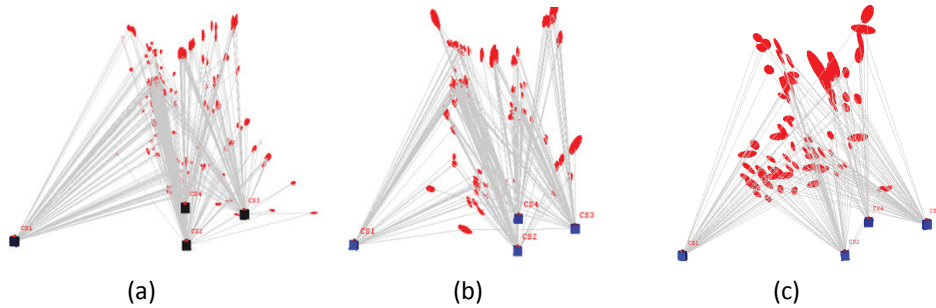


Fig. 5.23: Comparison of error ellipsoids. (a) SIFT matching. (b) SURF matching. (c) NCC matching.

Table 5.2 shows a comparison between the three different matching operators for the sample data set of Fig. 5.22.

Table 5.2: Comparison between different guided matching operators.

	Guided matching		
	SIFT	SURF	NCC
No. of oriented images	4	4	4
No. of tie points	256	144	103
Avg. point marking residuals [pixels]	0.79	0.92	2.07
Avg. point precisions [m]	0.011	0.013	0.030

All the operators are succeeded to guide the matching even with a wide baseline stereo pair with a privilege to SIFT.

Furthermore, 15 images of the same network are tested for the guided matching using the SIFT operator and compared to a commercial software (Photomodeler 2010). The proposed approach successfully orients the full dataset even with a wide baseline camera as shown in the red circle on Fig. 5.24a while Photomodeler was unable to orient the indicated camera.

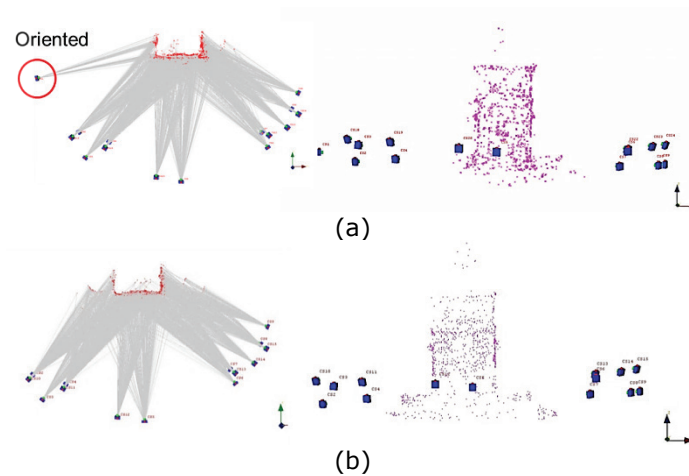


Fig.5.24: (a) Guided matching results in 15 oriented images. (b) Smart match (Photomodeler) results in 14 oriented images.

A comparison is also made between the proposed technique and the Photomodeler smart match in terms of: number of oriented images, number of tie points, marking residual, and the point precision as illustrated in Fig. 5.25.

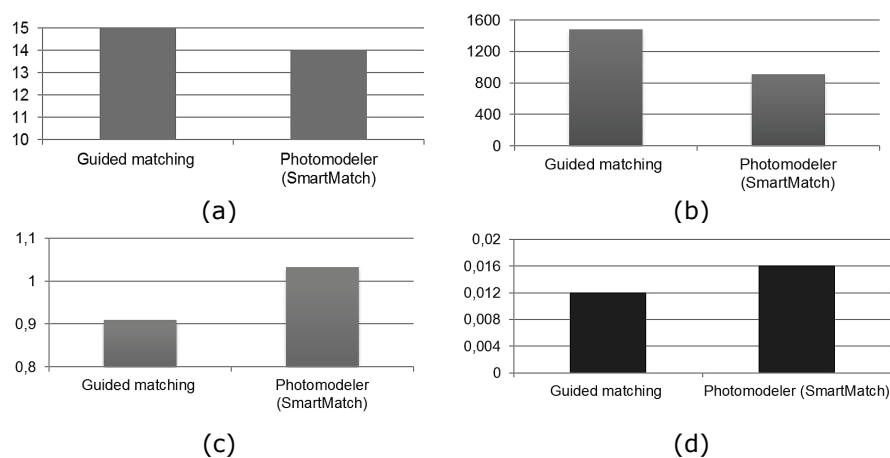


Fig. 5.25: (a) No. of oriented images. (b) No. of tie points. (c) The average point marking residuals [pixels]. (d) Average point precisions [m].

The graphs of Fig. 5.24 shows a promising results of the guided matching compared to Photomodeler.

5.4 Summary

In this chapter, we introduced a concept for guiding the camera operator to capture, in reality, the pre-designed optimal images presented in chapter 4.

The concept is based on creating synthetic images that visually represent the optimal camera orientation. This is motivated by the simplicity of interpreting the images to enable, even non-professionals, to accomplish the necessary photo take for the 3D modelling.

These synthetic images are created with a low resolution (like 640*480 pixels) by utilizing the texture of the captured video images that was used before for the optimization step. This selected resolution proved to be sufficient in the way that it gives enough SIFT matching points and on the other hand, it offers a reduced processing time needed when creating these images and applying the matching.

However, SIFT matching with synthetic images can result in a significant amount of wrong matches, which must be filtered to end with an efficient orientation. The method is based on simple statistical testing to remove these blunders with respect to a certain threshold (section 5.1.3).

A strategy is presented to accept or reject the real-captured images by validating the accuracy after incorporating the computed orientation of these images in the imaging network instead of those of the synthesized images. This is accomplished image by image until finishing the image capture for the whole designed network (Fig. 5.12). The accuracy validation is done in the same way as introduced in chapter 4 by computing the covariance matrix of the coordinates (Q_s).

The taken images are oriented by first applying the tie point matching. The proposed methodology was to guide the connectivity between the images instead of the time consuming full pairwise approach as shown in Fig. 5.21 which compared the processing time in both approaches.

The presented approach showed that the tie point detection is constrained to the object itself without its surroundings or the background (Fig.5.7) because the method is based on the derived rough 3D model of the object. This restriction limited the search space between the stereo images, avoided the mismatching points resulted from the dense texture in the background, and reduced the processing time. The approach proved, as shown in the experiment, to give better results in terms of marking residuals, number of oriented images, and final error (Fig. 5.25).

The advantages of using the multi-window matching procedure are concluded in the sense of:

- The avoidance of mismatching the SIFT correspondences due to the high probability of matching the same overlapped subimages.
- The large number of correspondences that will be computed with respect to the selected window size.

- The possibility to have a successful correspondence in case of wide baseline configurations as will be shown in chapter 7.
- The reduced consumption of computer memory in multi window matching when compared to the required memory of the full resolution images. Consequently, this will reduce the processing time for the correspondences matching.

-
However, some difficulties must be indicated in this developed guided imaging method. The first issue is concerned with the synthetic images where sometimes they lack the necessary texture. This is due to the incompleteness of the developed 3D rough model. Another problem is possible when the matching SIFT points are not well distributed around the image. This will affect the efficiency of results if applying a camera self-calibration.

Chapter 6 **Gap Detection and Final Modelling**

The image based techniques of camera planning and the image orientation described previously in chapter 4 and chapter 5 ends with a dense point cloud of the imaged object. However, the point cloud can still have some gaps despite the good camera planning because of different factors like the occlusion effects and the lack of texture on some object surfaces.

Generally, gaps in the final 3D image-based representation of objects can be decreased but barely avoided. This is due to several reasons that belong to the object complexity, object's surroundings and accessibility, sensor planning and the possible shortcoming in post-processing.

In the context of gap detection, post-processing techniques are applied based on either computer graphics or computer vision techniques. In computer graphics, gaps are to be filled graphically based on volumetric or surface meshing techniques. In computer vision, reducing the gaps is based on the improvement of the state-of-the-art stereo dense image matching. However, these processes are sometimes insufficient and can be assisted, as proposed in this chapter, by improving the imaging plan to offer better coverage of the missing parts of the modelled object.

Practically, this can be done by the automatic detection of the gaps and then by the addition of auxiliary image shots to the pre-captured image network. This will contribute in filling, as far as possible, the detected gaps.

Hence, this chapter will discuss the gap detection in a point cloud and the subsequent image capturing for the final 3D modelling.

6.1 *Gaps possible causes*

Gaps in the final image based 3D model (mainly with buildings) can result from different causes during the acquisition. These causes can be classified as:

- Insufficient object coverage by images.
- Structural and/or textural complexity of the object.
- Occluding objects between the imaging camera and the object.
- Poorly textured parts of the object.
- Existence of real holes or openings in the object.

These causes are illustrated as follows:

6.1.1 The insufficient object coverage by images

As presented previously in chapter 4, we designed the camera network in a way that guarantees full coverage of the study object. However, this design is based on a rough model of the object. If parts are missing in this model, they will possibly also be missed in the final imaging network and hence result in a gap in the final point cloud. Furthermore, our design is focused on taking the images from the street level, which is applicable particularly for laymen. This street level imaging is definitely a source of insufficient coverage because the upper parts of the objects like buildings or facades will be occluded as illustrated in Fig. 6.1.

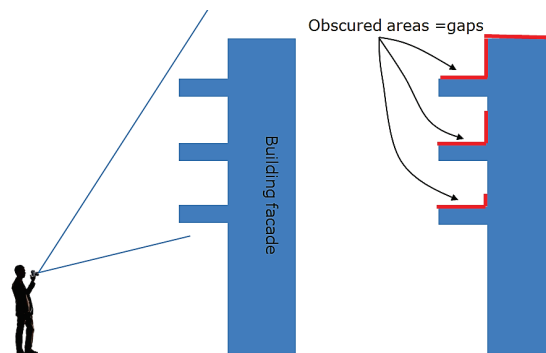


Fig.6.1: Obscured areas from street level view.

6.1.2 The complexity of the object itself

In the sense of 3D modelling, objects like buildings or statues are representing challenging (complex) objects in most of the cases. For sculptures and statues, the non-uniformity represents a complexity. On the other hand, building façades with their furniture (doors, windows, protrusions, etc.) are also complex to be modelled in a reality-based representation (Guidi et al. ; Manferdini and Remondino 2010). These kind of complexities is a major source of gaps in the 3D model. We classified these causes of gaps as follows (Fig.6.2):

- Gaps caused by self-occlusions
- Gaps caused by intrusions and extrusions (protrusions)

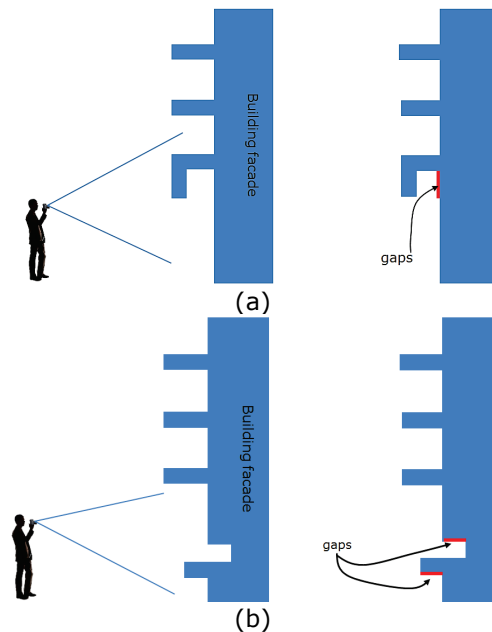


Fig.6.2: Gaps caused by object complexity a) Gaps due to self-occlusions. (b) Gaps due to protrusions.

6.1.3 The occluding objects between the camera and the object

The occluding objects between the camera and the study object represent a very common, difficult to avoid source for gaps in the final 3D models. Objects like vehicles, trees, fences, etc. are always preventing the free-occlusion 3D models. Many researchers have tried to solve this problem (Shin et al. 2008; Lee et al. 2010; Hammoudi et al. 2012). However, it's still a very challenging topic in image processing, photogrammetry and computer vision. The created rough model in the pre-planning step (chapter 4) will certainly suffer from this occlusion effect and consequently affect the designed camera network. Fig. 6.3 illustrates how the gaps can be caused by occlusions.

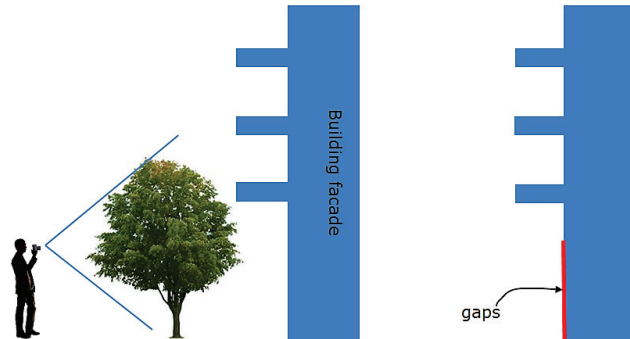


Fig. 6.3: Gaps due to occlusions.

6.1.4 The poorly textured parts of the object

Textureless objects like window glass is a common source of gaps when modelling building facades (Fig. 6.4). In image based techniques windows can be partly modelled due to shadows or curtains, but mostly they are incompletely modelled. There are also other parts which might cause problems, i.e. homogeneously painted façades.

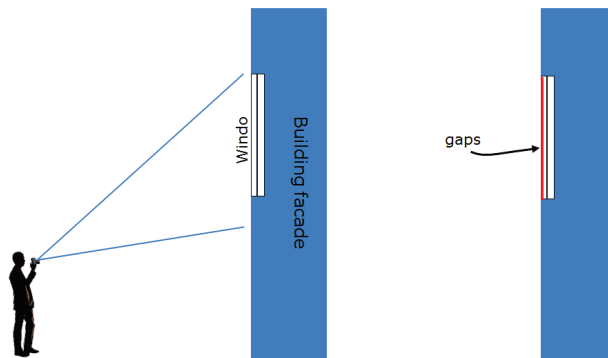


Fig. 6.4: Gaps due to textureless objects.

6.1.5 The existence of real holes (openings) in the object

In many building facades, a real opening can be found as a part of the architectural design of the building. Open gates and corridors are also common within facades. This is naturally represented as a gap in the final model as illustrated in Fig. 6.5.

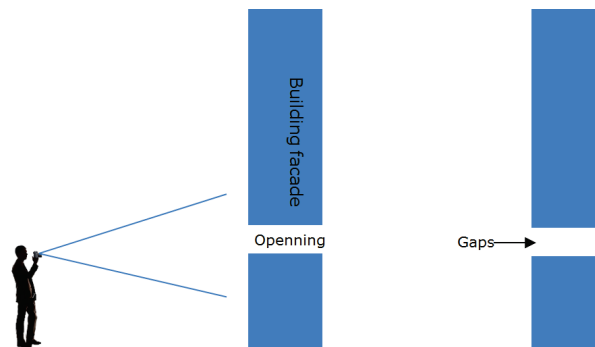


Fig. 6.5: Gaps due to openings.

Other minor sources for gaps can be: the unfavorable radiometric characteristics of the images due to illumination conditions and camera settings. Moreover, gaps can also be resulted from the deficiency or imperfection in the dense matching approach (Huq et al. 2013).

The following examples are prepared to illustrate the aforementioned causes of gaps in the final image based model. All the tests are implemented with a Canon 500D SLR camera with 18mm focal length by using state - of -the -art Agisoft software (Photoscan 2011).

Example 1

High resolution images are taken for a building façade from a street view where a car is parked beside the building as shown in Fig. 6.6. The four images are oriented automatically and the final textured point cloud is shown as well.

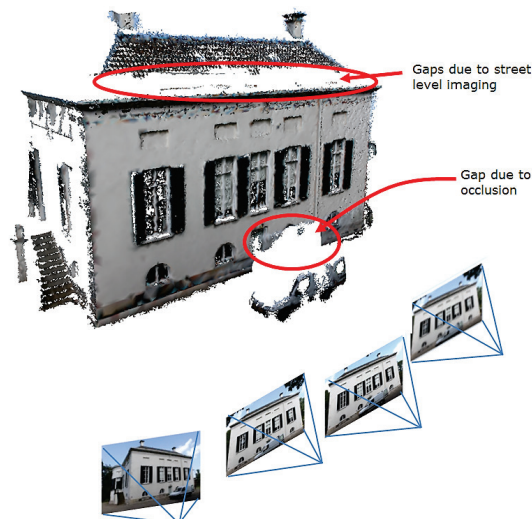


Fig. 6.6: The effect of occlusions on the completeness of the final 3d point cloud.

Example 2

This example applied with 17 images taken to a building which has a more complex architecture as shown in Fig. 6.7. The resulted 3D model contains few gaps due to the textureless window glass and the openings of corridors. The gaps near the roofs are also expected because of the street level view and self occlusion effect.

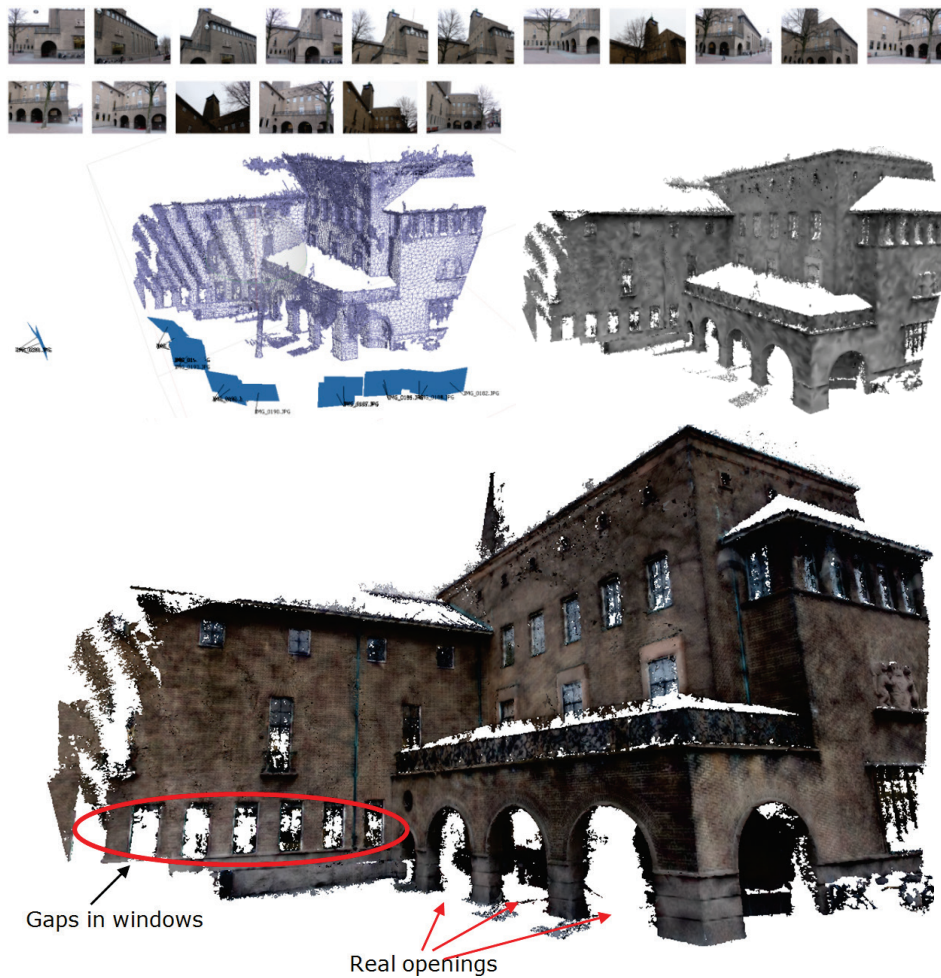


Fig. 6.7: The effect of textureless parts, openings and self occlusion on the completeness of the final 3D point cloud.

Example 3

To illustrate the gaps resulted from protrusions, we selected five images of a façade with columns shown in Fig. 6.8. The resulted 3D model contains gaps along the column sides due to the effect of protrusions and image viewing directions.

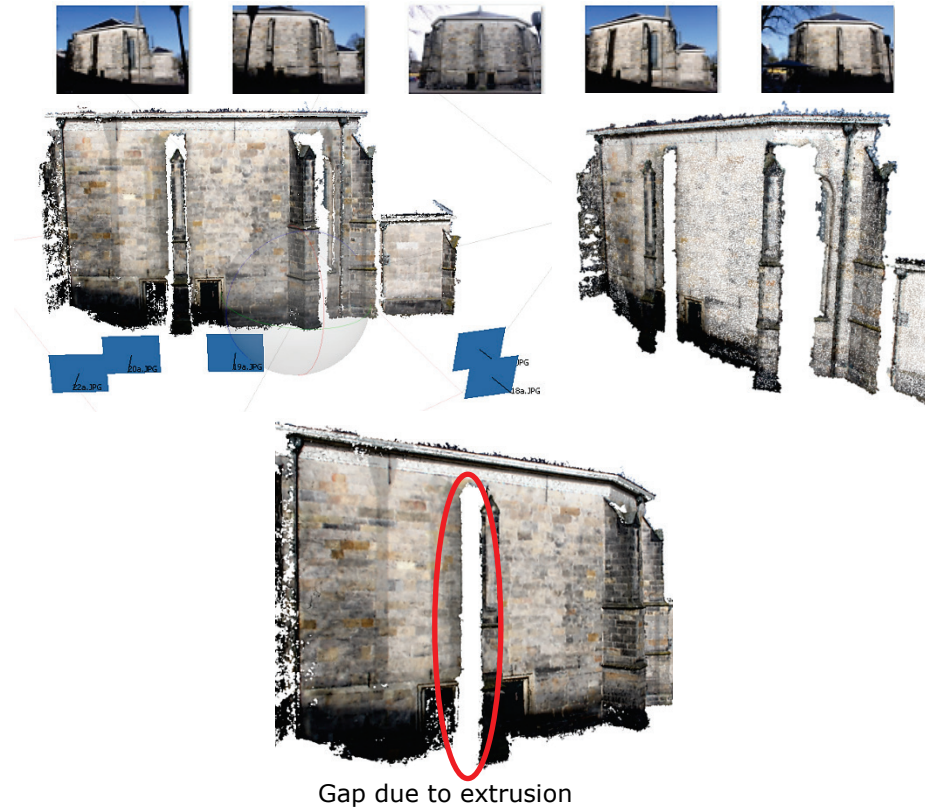


Fig.6.8: The effect of protrusions on the completeness of the final 3d point cloud.

In this thesis chapter, we are mainly focusing on the detection and recovery of the point cloud gaps resulted from the insufficient camera planning. The avoidance and treatment of gaps resulted from unavoidable occlusions like building roofs, and poor texture areas will not be addressed here.

6.2 Gaps detection techniques

As stated previously, two methods of gap detection are proposed in this chapter. The first method which is computationally less expensive is based on a 2D space representation and processing, while the second method is based on the 3D voxelization of the object space. The voxelization method is computationally expensive, but more effective, especially if real openings exist in the point cloud.

6.2.1 2D-based processing

Possible causes for gaps in the image-based point clouds are discussed earlier in section 6.1 and the question is how to detect these gaps with an

automated, simple and accurate approach. This step will benefit from the previous steps of camera planning and image orientation of chapter 5 where each image in the network is accurately oriented.

On the other hand, the proposed technique is not counting on any created surface mesh. This should be an advantage in the sense of avoiding expensive computations and possible errors or insufficiency in the surface meshing. Fig. 6.9 shows two examples of triangulated meshes where the mesh failed to model the real openings in Fig. 6.9a or the occluded gap of the columns in Fig. 6.9b. Therefore, the technique might be inefficient because it is based on detecting elongated mesh triangle sides (Impoco et al. 2004) which is not possible to be modelled in some cases as shown.

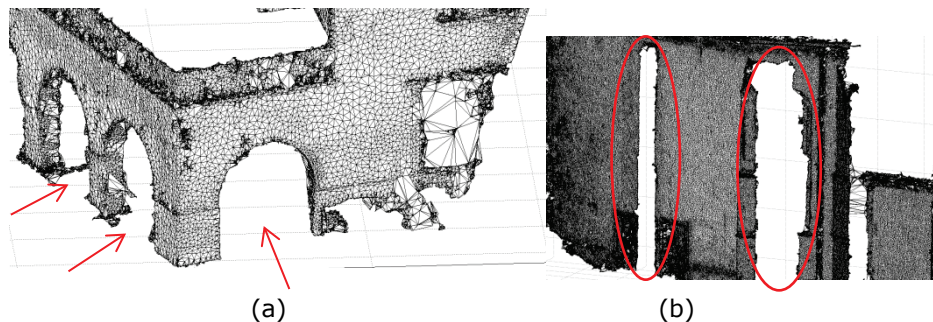


Fig. 6.9: Triangulated surface mesh for gap detection. (a) Building with openings. (b) Facade with occluded gaps.

Therefore, we propose the 2D based gap detection, which is summarized as follows:

1. The projection of the point cloud back to the source images. This is applicable since the image orientations beside the point cloud are well computed in the step of image orientation.
2. The projected points are to be represented as a raster format. For simplicity, the images are created in a binary format (BW) as shown in Fig. 6.10b.
3. Morphological operation is applied by using the image morphology closing. Image dilation is first used to clear the null value (black) of pixels within the point cloud outline and to replace them with filled pixels (white). The image morphology closing is preferably applied with a 'line' structural element (SE). A line SE is more efficient to preserve the outline of the objects like buildings as shown in Fig.6.10d. The size of the SE is critical to have accurate detection and it is based on the density of the point cloud and the gap size.

For our tests, the raster image size is set to 480×320 pixels and the line SE to 30 pixels. The improper selection of SE size is shown in Fig.6.10c where the point cloud boundaries in the upper part are extended wrongly to image boundaries. However, after the closing operation, the image boundaries are filled with a 10 pixel strip of null values to ascertain accurate detection.

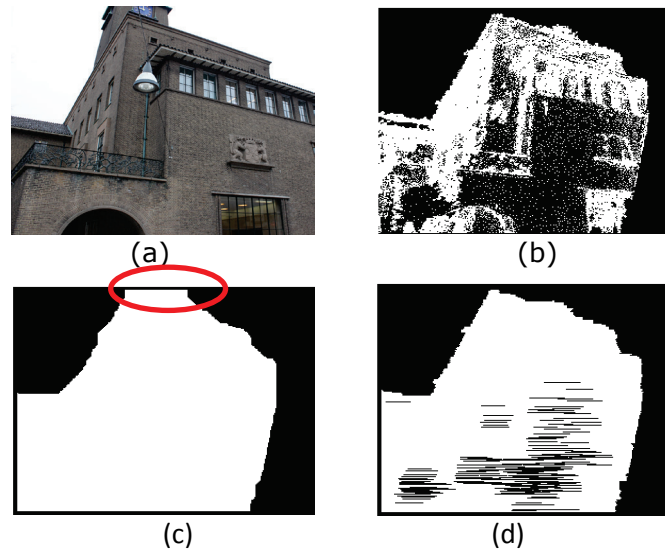


Fig. 6.10: Image morphology on the raster image of the projected point cloud. (a) Original image. (b) BW image of the PC. (c) Image closing with disk SE. (d) Image closing with line SE.

4. Accordingly, the original BW image is subtracted from the dilated image to detect the possible gaps inside the object outline. This subtraction is important to isolate the gaps from the image background. Due to the varying density in the point cloud, distinct pixels might be assigned as gaps as well.
5. To avoid the gaps resulted from the subtraction in the previous step, a smoothing image filter (averaging filter) is applied to the image. The filter size as the SE size is proportional to the image resolution. This is because whenever the resolution increased, the density (proximity) of the point cloud in the raster image increased which needs a larger filter size. However, for our tests, a filter size of 7×7 pixels is used.

Fig.6.11 illustrates the methodology workflow. It must be mentioned that this technique can be efficient when the point cloud is dense. Otherwise sparsity can affect the results. Moreover, this method is expected to be insensitive to small size gaps. The methodology is to be applied for every source image and then the final gaps are defined in the 3D space by exploiting the predetermined 3D point cloud form the previous steps (chapter 5).

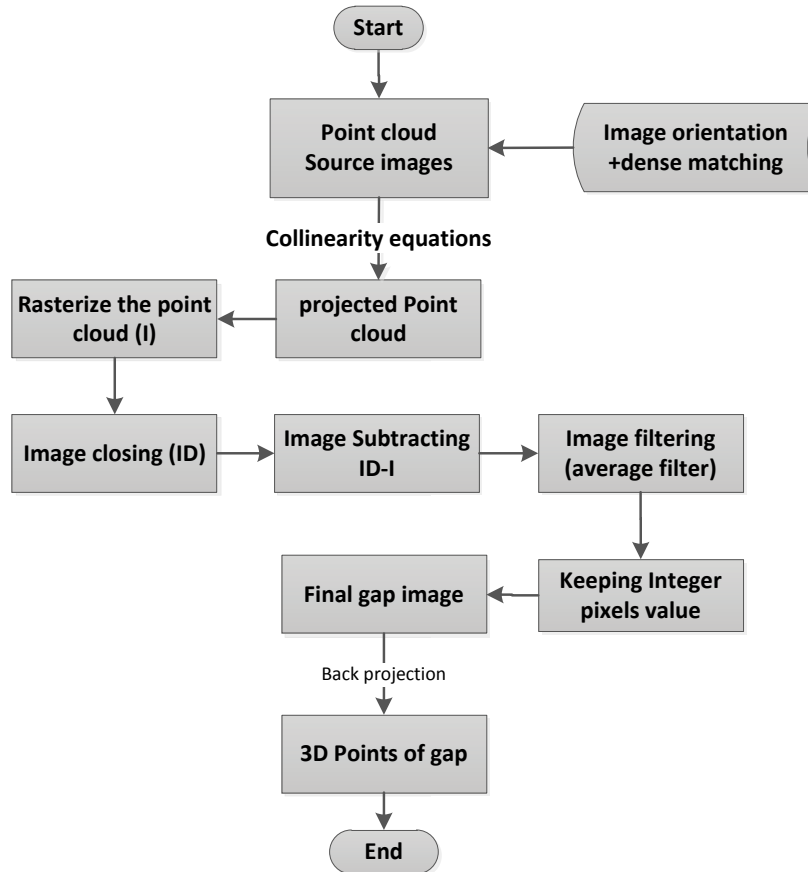


Fig.6.11: Methodology of 2D-based gap detection.

The same dataset of Fig.6.8 of a facade point cloud produced from five images will be processed for the gap detection as illustrated in Fig.6.12.

Fig.6.12b shows six successive steps for processing the point cloud of the facade to determine the gaps. Starting from the raster image of the point cloud, a morphological closing is applied followed by subtraction. Then a smoothing filter is used and followed finally by filtering to reach the final detected gap.

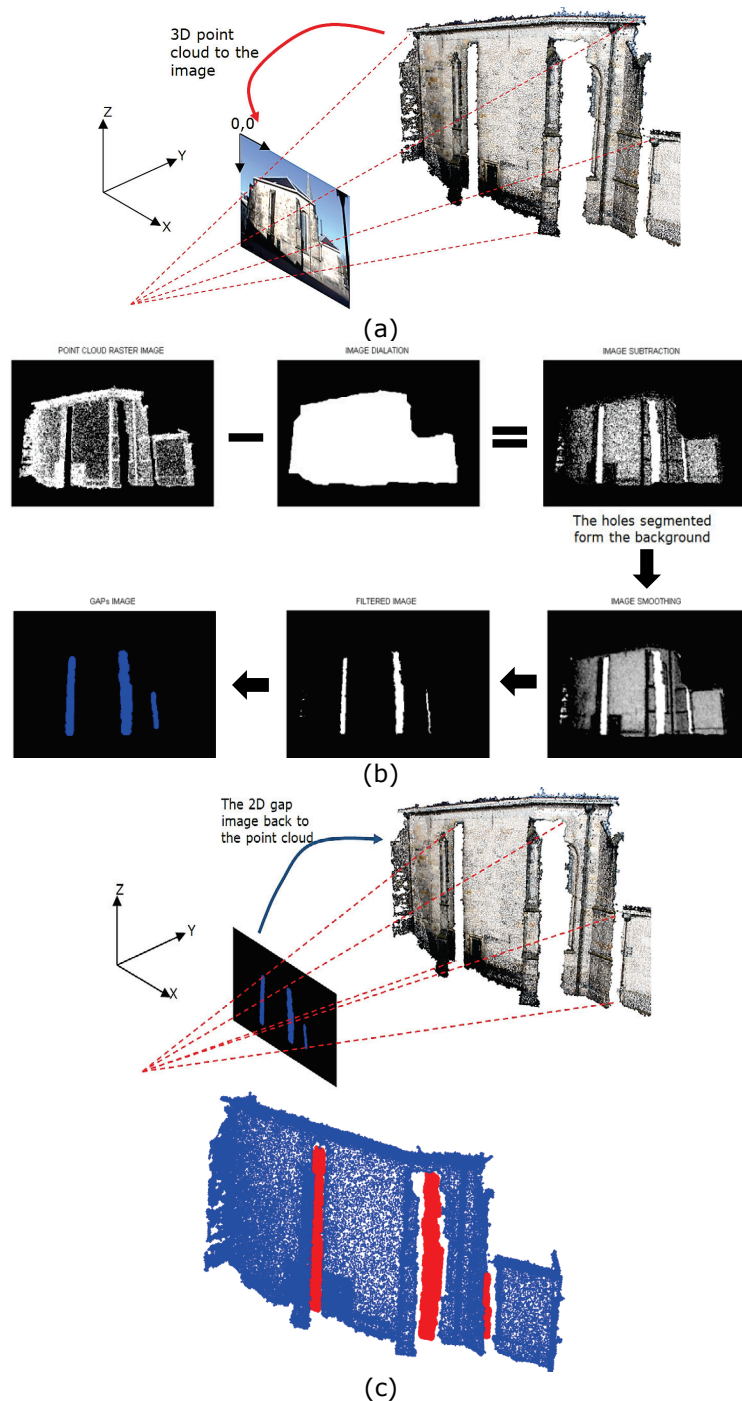


Fig.6.12: The 2D-based gap detection experiment. (a) Projecting the point cloud to the images. (b) Gap detection technique. (c) The 3d projected gap points.

Another experiment of a building façade covered by 14 images is shown in Fig.6.13. The dense point cloud is created by using the Agisoft software. A car parking near the building caused a gap in one of the stair sides.



Fig.6.13: Façade dense point cloud.

An image view is selected and processed for the 2D gap detection as shown in Fig.6.14a with a structural element size of 30 pixels.

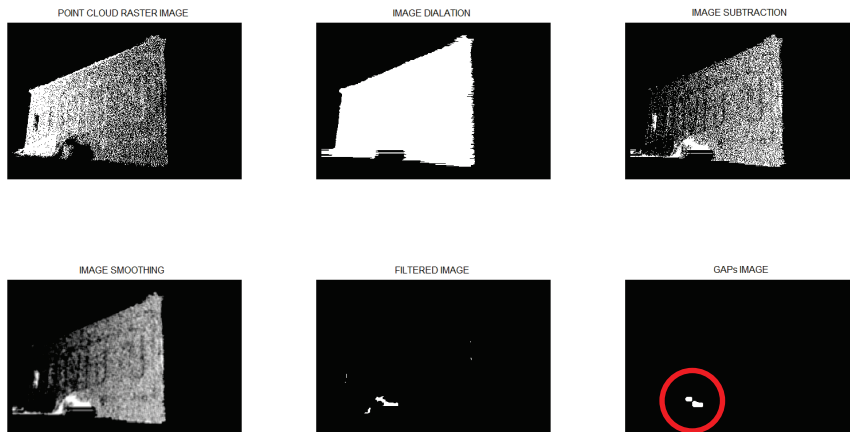


Fig. 6.14: The gap detection in a building stairs.

The discussed 2D-based gap detection technique has the disadvantage of missing the 3D depth perception in some cases when openings are existing. This is because it is based on projecting the point cloud back to the image planes. Furthermore, the edges of objects like facades are sometimes, mistakenly, considered as gaps due to the morphological dilation or the improper selection of the structure element (SE) shape and size. Therefore, the second proposed technique for gap detection is based on the 3D space representation.

6.2.2 Gap detection by voxelization

The detection of gaps in 3D space can be accomplished either by using a volumetric representation with voxels or with triangulated surface meshing. Volumetric space representation has the advantage of avoiding the intermediate step of triangulation which might be a source of erroneous detection.

Hence, the following 3D based methodology of Fig.6.15 is developed for the detection of the gaps in the image based point cloud.

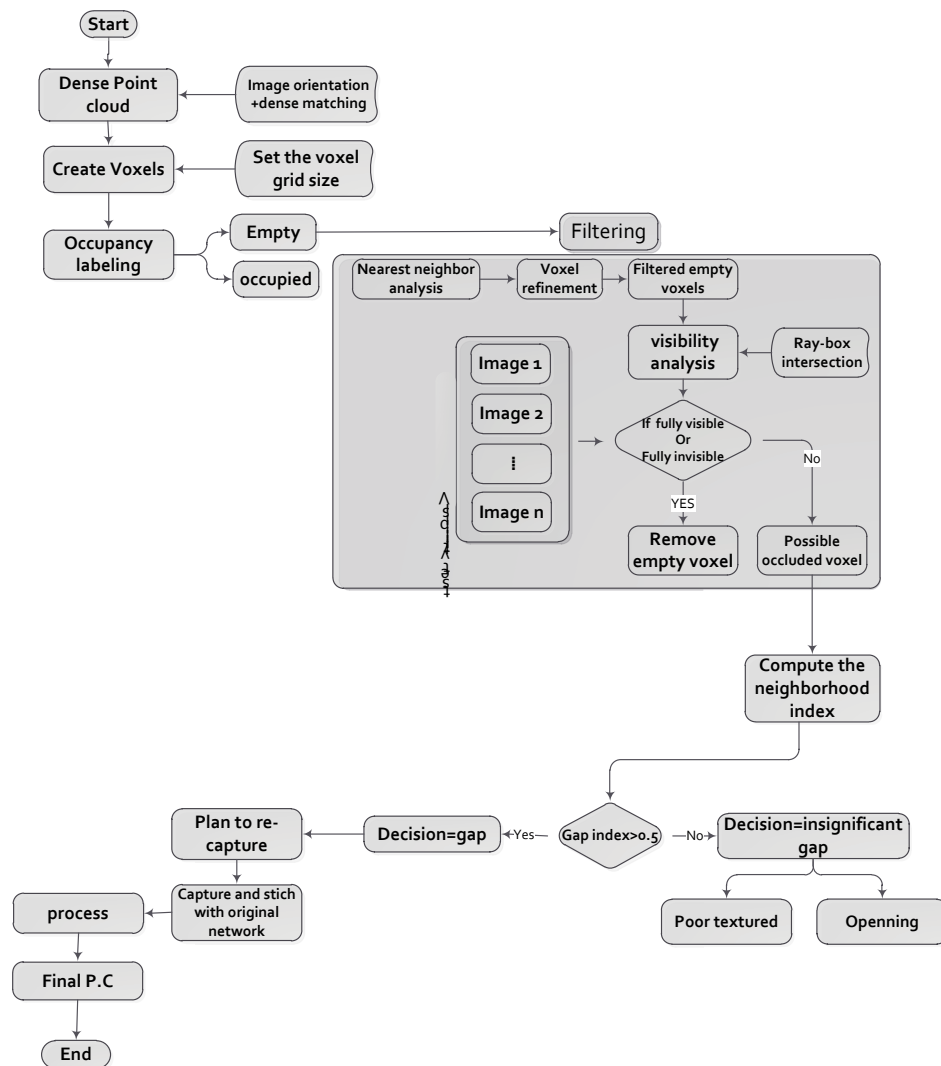


Fig. 6.15: The workflow of the 3D based gap detection.

For more details, the steps of gap detection are explained as follows:

a) Voxelization

Construct the voxels in 3D space that span the space of the point cloud. Each voxel will be labelled either as empty or occupied based on the point cloud as shown in Fig.6.16.

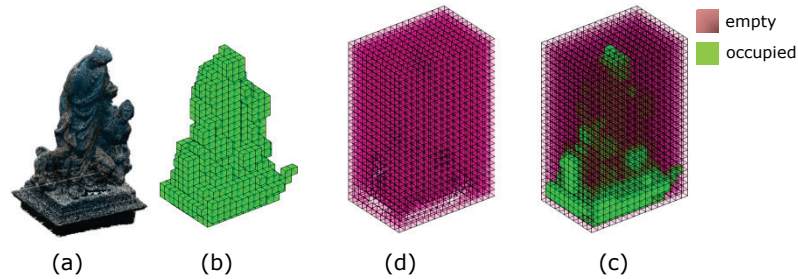


Fig.6.16: The 3D voxelization of point cloud with 20 cm voxel side. (a) Point cloud. (b) Occupied voxels. (c) Empty voxels. (d) Combined visualization of the empty and occupied voxels.

b) Empty Voxel refinement

Initial filtering of the insignificant or blundered empty voxels is to be done by keeping the nearest empty voxels to the occupied voxels. The refinement is done by using the occupied voxels to glue the neighbouring empty voxels and discard the others. The neighbouring empty voxels are found by computing the 'nearest neighbour' with a $3 \times 3 \times 3$ buffer filter (Fig.6.17). This is followed by removing the distant empty voxels which represent the marginal or bordering voxels away from the occupied voxels of the point cloud.

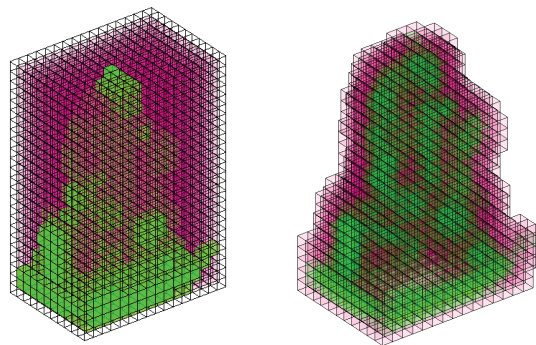


Fig.6.17: The initial filtering of voxels. (a) Before refinement (b) After refinement.

c) Visibility analysis

A second step of voxel labelling and filtering is based on the visibility analysis between the viewpoints and the voxelized point cloud. This analysis can be prepared by using line-voxel intersection or line tracing method as discussed in chapter 2. Three types of labeling to the empty voxels might be found as

illustrated in Fig.6.18 and Fig.6.19: fully visible, partially occluded, and fully occluded voxels.

Based on the assumption of a sufficient camera network, fully visible and fully occluded empty voxels will be neglected. For partly occluded voxels, more investigation is required to look for potential gaps. Neglecting fully visible voxels is based on the fact that empty voxel cannot occlude occupied voxels (Fig.6.18a), and thus must actually be empty. Furthermore, real openings like open doors can also be modelled with fully visible empty voxels and will be neglected (Fig. 6.18b).

Furthermore, fully hidden occluded voxels like empty space behind a wall are also neglected as illustrated in Fig. 6.18c.

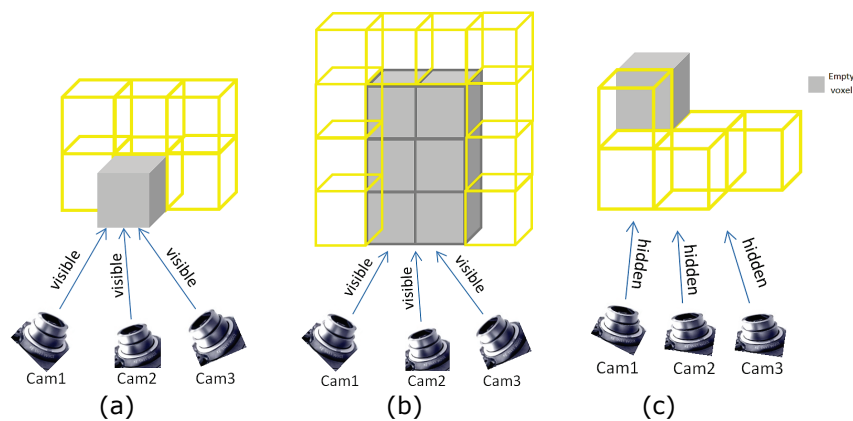


Fig.6.18: Labelling empty voxels by visibility analysis. (a) Fully occluded. (b) Real opening. (c) Fully visible empty voxel.

On the other hand, partly occluded empty voxels can be considered as gaps because either they are not covered well by the images (less than three images) or because they are occluded by another object (Fig.6.19).

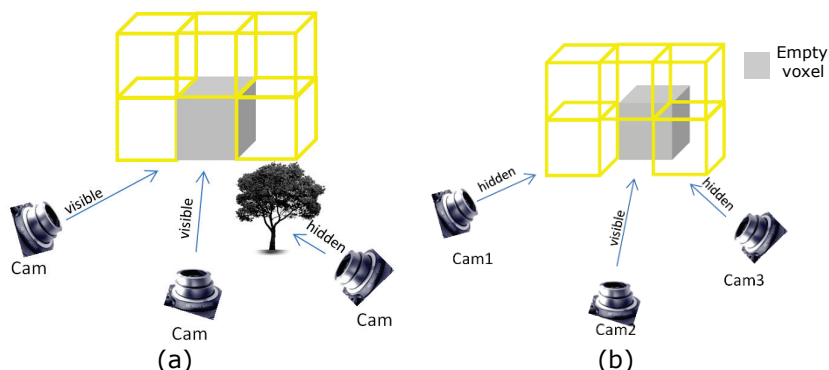


Fig.6.19: Labelling partly occluded empty voxels by visibility analysis. (a) Obstacle-occlusion effect. (b) Insufficient covering cameras or self-occlusion.

The labelling the empty voxels as fully occluded, fully visible or partly occluded can be expressed in the following pseudo code in Table 6.1.

Table 6.1: Pseudo code for the visibility analysis of empty voxels.

```
For every empty voxel  $A_j$ 
For every image  $i$ 
origin=  $[Tx_i, Ty_i, Tz_i]$ ,
direction=  $A_j$ -origin,
Compute distance between the empty voxel  $j$  and camera  $i$ :  $d_{ij}$ 
Compute rotation matrix  $M$ ,
Compute image coordinates  $x_j, y_j$  by collinearity,
%%Check if the voxel is within the camera view
if( $x < \text{length}/2$  &&  $x > -\text{length}/2$  &&  $y < \text{width}/2$  &&  $y > -\text{width}/2$ );
%% then check if it is occluded or not by the other occupied voxels
for every occupied voxel  $B_k$ 
define the two vertices of a voxel  $k$  ( $v_{min}, v_{max}$ ),
flag = rayBoxIntersection(origin, direction,  $v_{min}$ ,  $v_{max}$ )
Compute distance between the occ. Voxel  $k$  and camera  $i$ :  $d_{ik}$ 
if flag==0 there is no intersection
F( $k, l$ )= 0
else if flag==1 &&  $d_{ij} < d_{ik}$  then the occ. Voxel  $k$  is behind the empty voxel  $j$ 
F( $k, l$ )=0 ; %% and the empty voxel is visible.
else if flag==1 &&  $d_{ik} < d_{ij}$  then the occ. Voxel  $k$  is in front of the empty
voxel  $j$  and closer to camera  $i$ ,
F( $k, l$ ) =-1; %%the empty voxel is invisible
%%
Repeat for every occ. voxel
Check if sum(F)=0, then the empty voxel  $j$  is visible;
Check if sum(F)~=0, then the empty voxel  $j$  is invisible;
Repeat for every camera  $i$ 
Repeat for every empty voxel  $j$ 
If voxel  $j$  is visible >2 cameras
Vindex=1
else if voxel  $j$  is hidden in all cameras
Vindex=-1
Else
Vindex=0 %% partly occluded
End
```

d) Neighbourhood index

The visibility analysis and labelling of empty voxels is not always sufficient to reliably find out the potential gaps. This is because of the possible inadequate camera placement and the blunders or noise in the point cloud. Moreover, openings modelled with empty voxels might, sometimes, not be visible from every imaging camera and will be labelled as partly occluded.

Therefore, other measures are to be added to support the visibility analysis. A neighbourhood index (NI) might be efficient to strengthen the detection of gap voxels. This is actually similar to a majority filter (3x3x3 neighbourhood), but also uses the actual direction of neighbours. Fig.6.20 illustrates the computation of the neighbourhood index of a voxel. Three types of proximity distances (d_1, d_2, d_3) are to be computed to define the search space. More neighbouring occupied voxels indicate a high chance of being an occluded

empty voxel and vice versa. NI is computed as (number of occupied voxels/ total neighbouring voxels) and a threshold of (>50%) is considered to indicate occluded empty voxel. On the other hand, the remaining voxels will be considered as insignificant gaps. These gaps might be blunders or caused by the textureless parts and real openings.

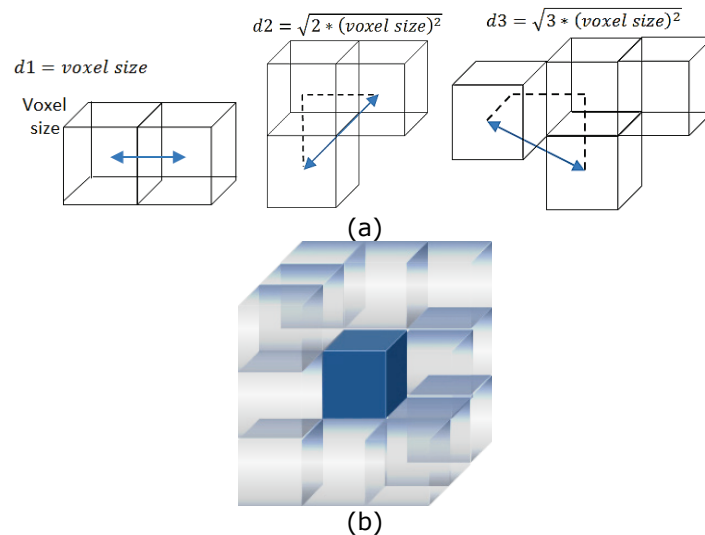


Fig.6.20: (a) Proximity measure. (b) 3x3 neighbouring voxels.

Finally, other measures like the altitude can be used to refine the detection. Altitude index is useful when the imaging is done with a street level view since the upper parts of the object are self-occluded. Hence, the empty voxels near the upper parts of objects will be neglected because of the visibility limitation.

Moreover, converting the images into binary can be used to assist the detection of real openings. These openings usually have a dark texture or low intensity values near zero and this might be useful to be added as an extra information as will be shown in the following experiments. The openings which are represented as black areas will probably need to be verified either as real dark spots or blunders to be removed. This detection is applied by successive image morphological closing. Black painted doors as an example will be considered as potential openings according to this methodology. However, these kind of objects is normally represented in the point cloud and later as occupied voxels and will not affect the detection.

After the gap detection, auxiliary images will be captured to recover the gaps in the point cloud and to finally have a complete 3D model.

Two experiments will be presented to verify the proposed 3D based gap detection by voxelization as follows:

Façade Experiment

The same experiment of Fig. 6.13 is tested with the algorithm of 3D gap voxelization which is applied based on a 30 cm user defined voxel size as shown in Fig. 6.21. Fig.6.21a shows the occupied voxels while Fig.6.21b shows the visibility analysis of the empty voxels. Fig.6.21c illustrates the partly occluded gap voxels and Fig.6.21d shows the final detected gaps after the neighbourhood filtering.

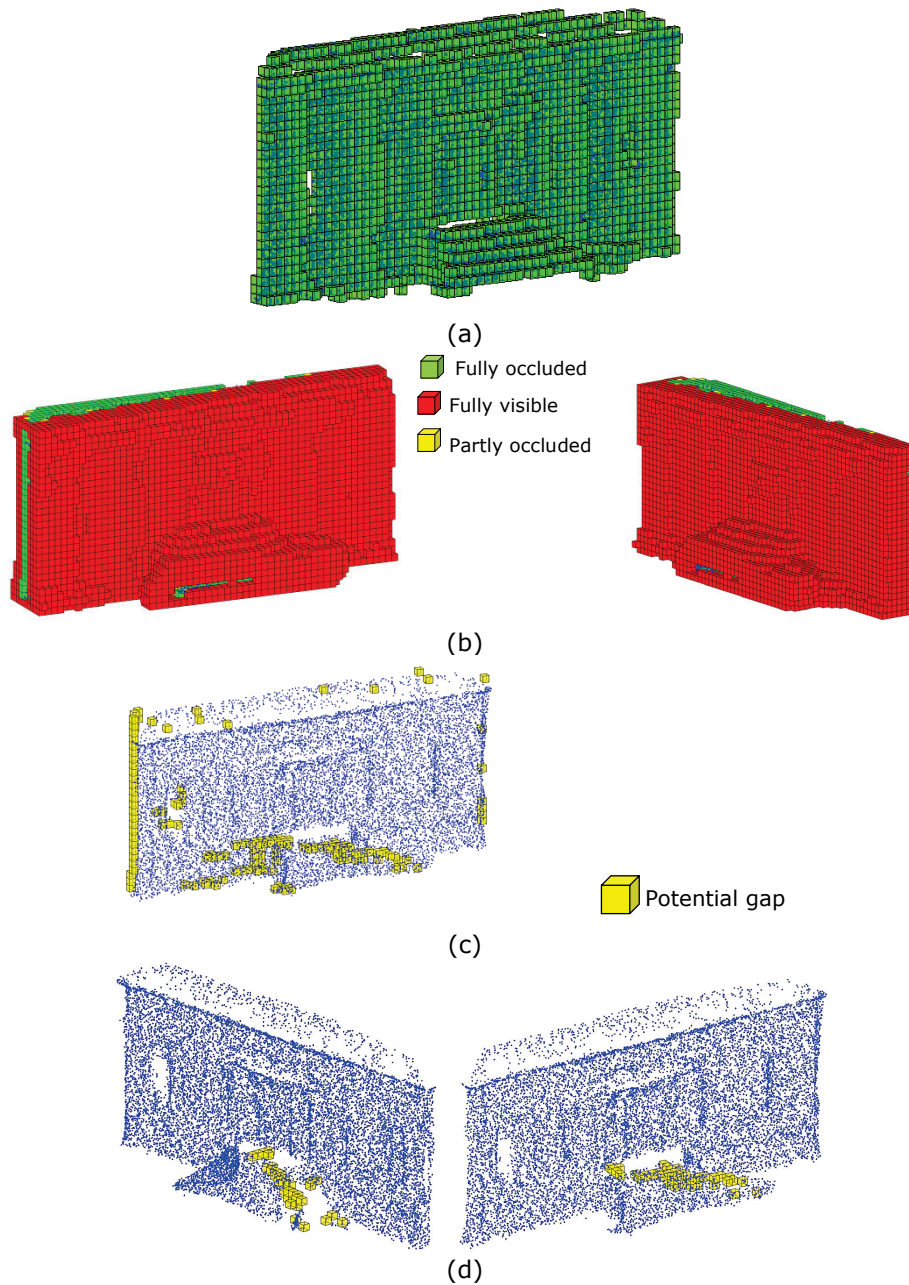


Fig.6.21: The output results of gap detection. (a) Occupied voxels. (b) Empty voxels labelled according to visibility. (c) Empty voxels After visibility test. (d) Final filtered empty occluded voxels.

The detection was successful to find out the gaps at the right side of the stairs as shown in Fig.6.21d.

Statue experiment

To evaluate the method on free form objects like statues, the 3D gap detection method is applied. 64 images are taken around a state as shown in Fig.6.22. The created dense point cloud is lacking some parts near the upper shoulder and near the rectangular base.



Fig. 6.22: Statue dense point cloud and gaps

The 3d voxelization gap detection is applied and the results are illustrated in Fig.6.23 based on user defined 15 cm voxel size. The developed method is successful to detect the empty voxels near the base regions and near the top shoulder of the statue, which is not well visible from the camera operator ground level (Fig. 6.24).

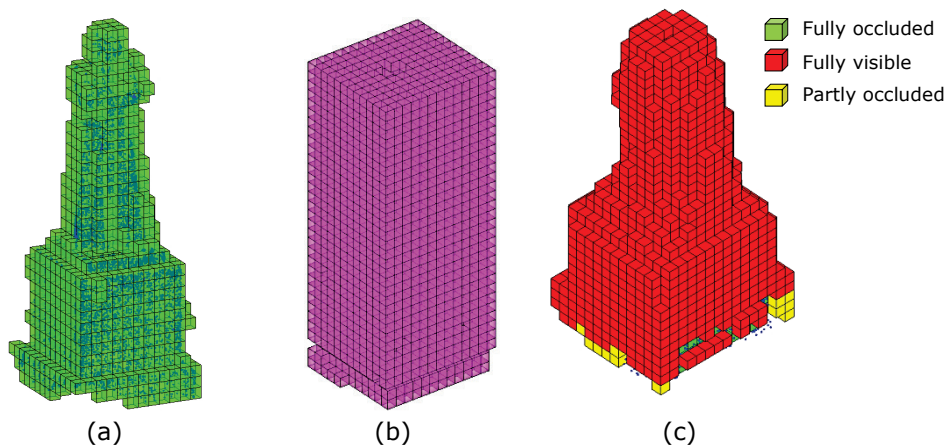


Fig. 6.23: Gap detection in a statue point cloud. (a) Occupied voxels. (b) Empty voxels and (c) Empty voxels after visibility test

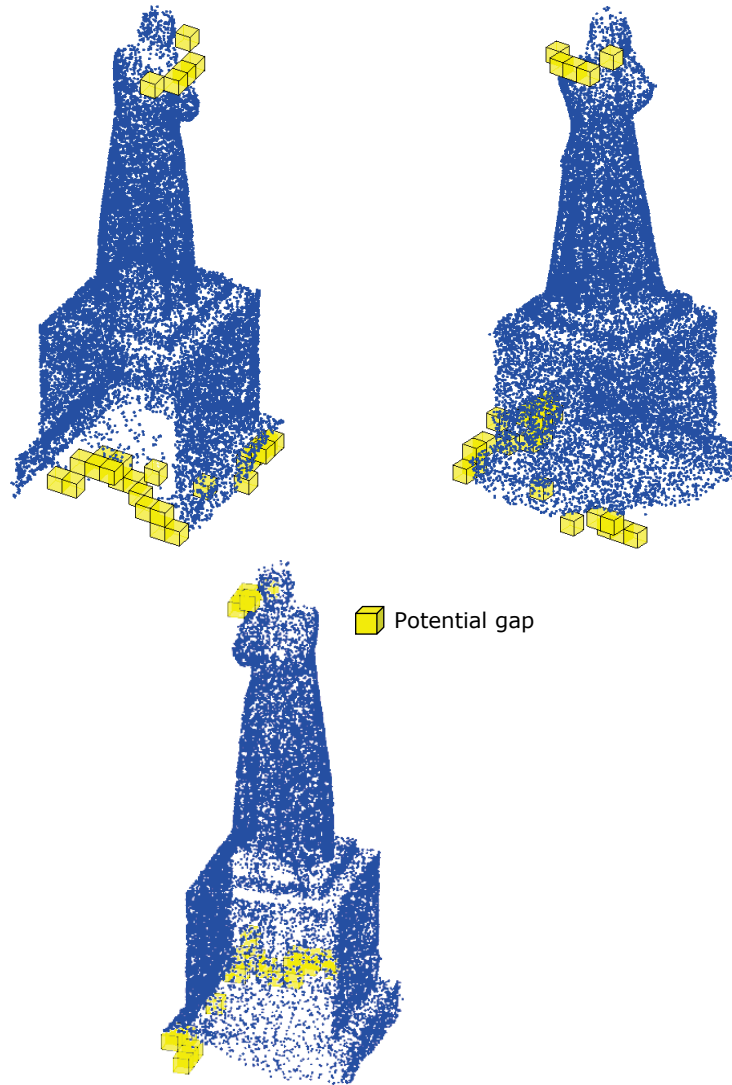


Fig.6.24: Gap detection in a statue point cloud

The detected voxel gaps will be avoided by taking auxiliary images for these gap areas as will be shown in section 6.3.

Facade with real opening experiments

The third test is applied on a façade with a real opening of a gate with a corridor as shown in Fig.6.25. Four images are selected to create a dense point cloud of the façade. The voxelization is applied similar to the first test and based on 30 cm voxel size.

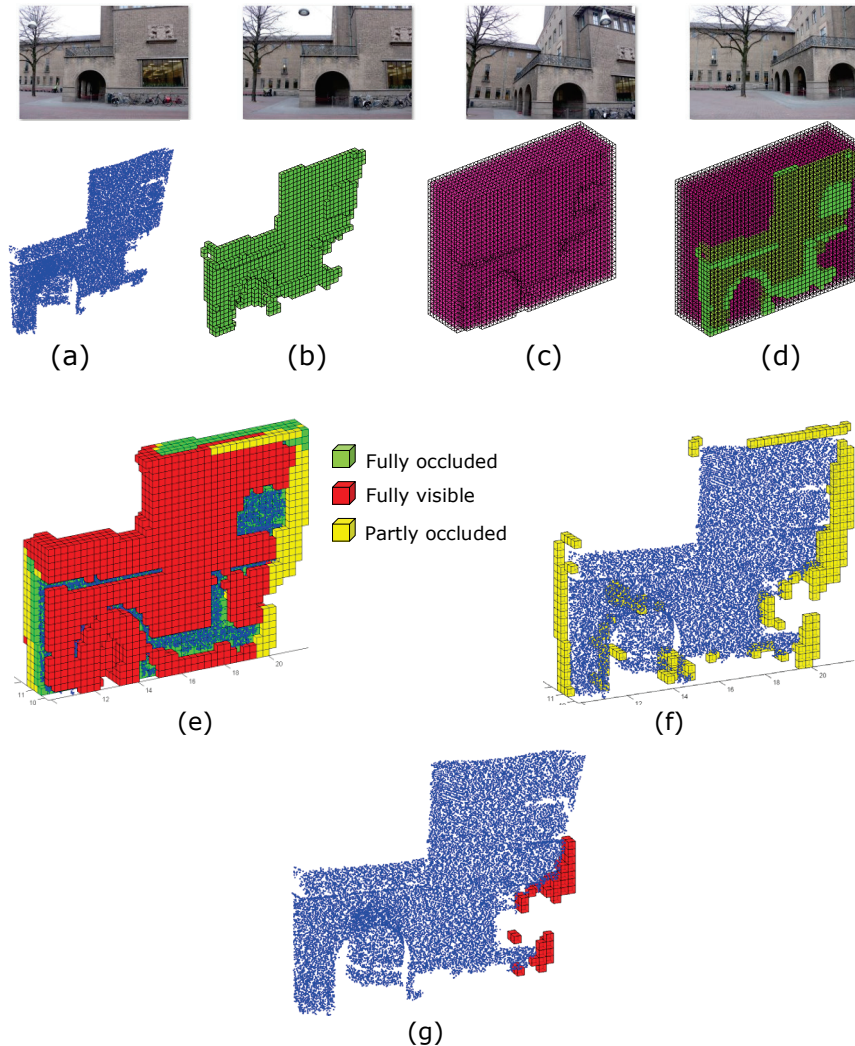


Fig. 6.25: Gap detection of a façade with opening. (a) Point cloud. (b) Occupied voxels. (c) Empty voxels. (d) Empty and occupied voxels. (e) Visibility labelling. (f) Partly occluded voxels. (g) Empty voxels after neighbourhood analysis and filtering.

Fig.6.25f shows the final detected gaps in red which are located in the missing part of a window in the point cloud. It must be noted that few voxels are detected inside the real opening after the visibility analysis. These voxels are detected because of the existence of some points behind the gate. However, these voxels are filtered out by the neighbourhood index.

A second test on a different façade of a garage is also implemented where six overlapped images are taken as shown in Fig.6.26a.

The point cloud shows the real opening of the garage beside some occluded gaps above the gates. It is expected to detect these occluded gaps while neglecting the empty space of the real openings of the garage.

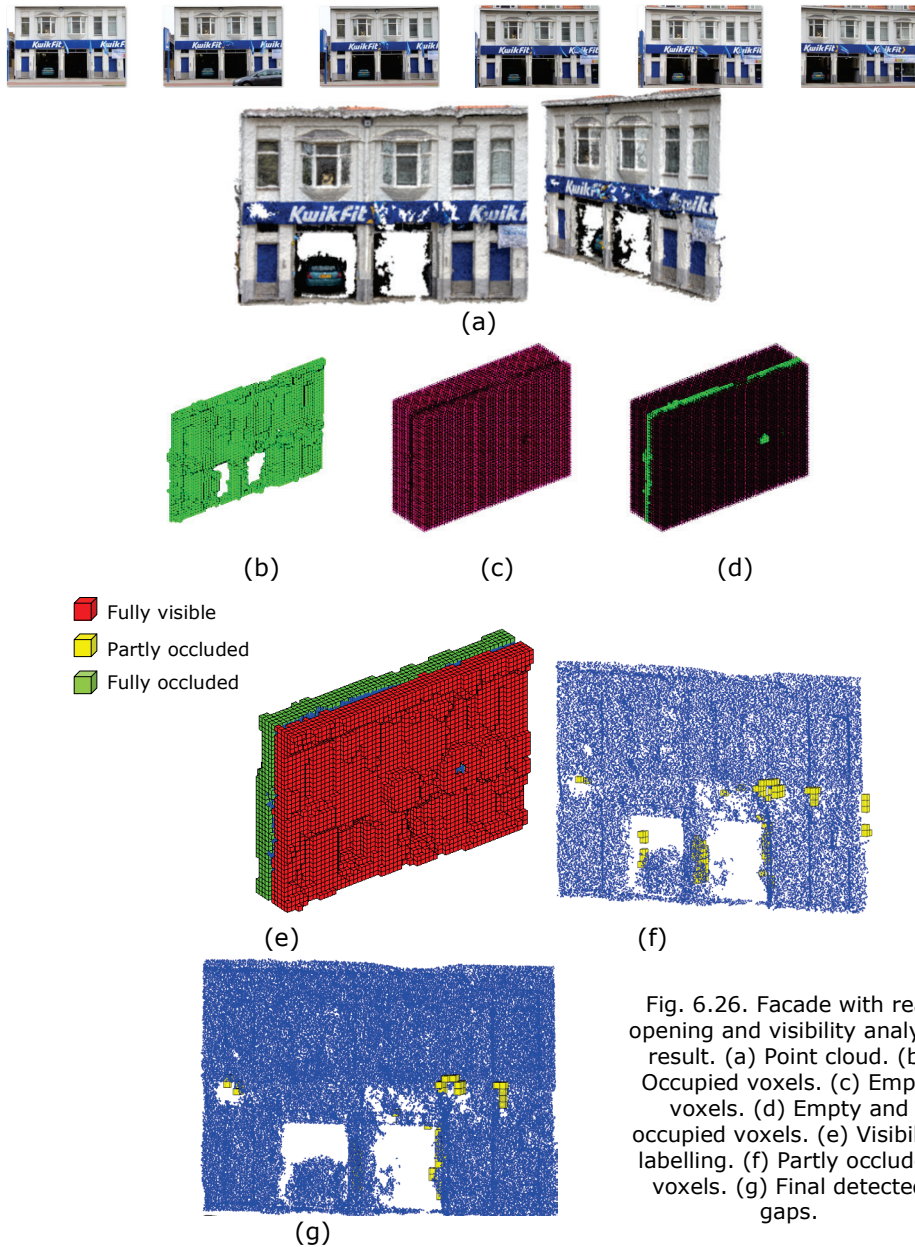


Fig. 6.26. Facade with real opening and visibility analysis result. (a) Point cloud. (b) Occupied voxels. (c) Empty voxels. (d) Empty and occupied voxels. (e) Visibility labelling. (f) Partly occluded voxels. (g) Final detected gaps.

After the visibility test of Fig.6.26f, some faulty detected empty voxels inside the openings are found. These wrongly detected voxels are partly occluded because they are located deeper from the façade plane because of the fill

approach and some noise in the point cloud. The use of neighbourhood index will remove these blundered voxels as shown in Fig. 6.26g.

6.3 The auxiliary image capture and orientation

After the detection of gaps in a point cloud, the planning on taking auxiliary images is followed as shown earlier in the methodology of Fig. 6.17. The concept of completing the detected gaps can be applied by projecting these detected gaps back to the source images and highlight their spots for subsequent guidance. Another idea that can fit the nonprofessional needs is by displaying the 3D model of the object with the gap spots or voxels highlighted on it. The user can easily trace and define these spots on the object and capture the auxiliary images without more guidance as before.

Three auxiliary images, at least, should be taken for every detected gap and the orientation step is applied again for the final step of dense point cloud and 3D modelling.

A more advanced procedure is by re-applying the camera planning described in chapter 4. The guidance with synthetic images can be applied based on the parts surrounding the detected gaps.

The same approach of guiding the connectivity between the images (chapter 5) will be implemented again to reduce the computations and to avoid the matching failure problem when object repetitive pattern exist.

The connectivity tree is simply computed between the previously oriented images involved with the detected gap area and the new auxiliary images. This is based on the assumption of a reasonable overlap between both sets of images. Fig. 6.27 shows a diagram of the imaging update for image connectivity.

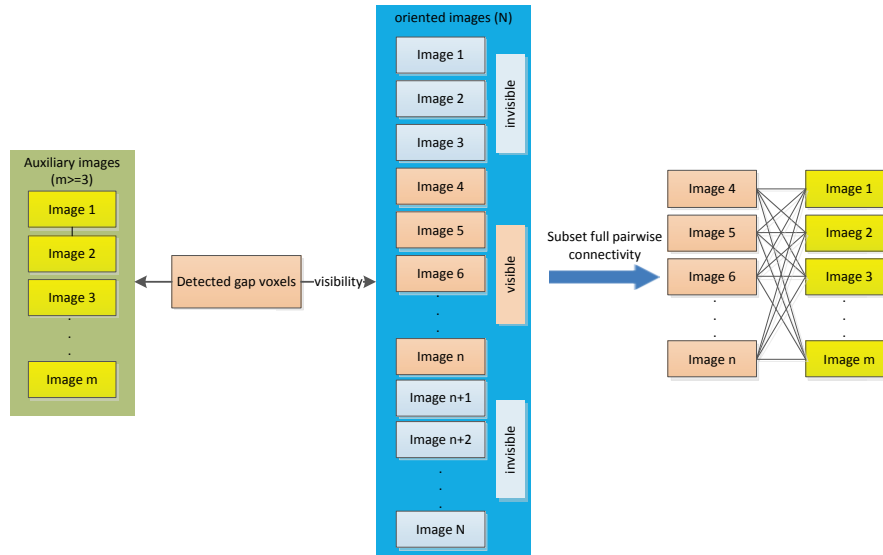


Fig.6.27: Image connectivity with auxiliary images

On the other hand, the guidance of SIFT matching will not be applicable as applied before (chapter 5) because of the missing 3D information in the detected gap areas.

The 3D illustration of Fig. 6.21 of the façade experiments will be useful to guide the camera operator to capture a minimum of three auxiliary images in the gap area. This is more convenient to be used with the current technology like with tablets or smart phone devices. Fig.6.28a shows the dense point cloud before gap detection and recovery. Fig.6.28b illustrates the dense point cloud (by Agisoft) after taking five auxiliary images that will be stitched with the pre-oriented images to complete the missing part of the stairs. Fig. 6.29 shows the dense point cloud of the statue base and the completeness after the addition of eight auxiliary images.

This proposed system can test the completeness of the point cloud and can check the gaps occurrence iteratively. It must be noted that the pose of the captured images will be more convenient at this stage to be estimated by the camera operator intuition without additional guidance. However, following some guidelines on image capturing or having some experience in photogrammetry is better to avoid the repetition.

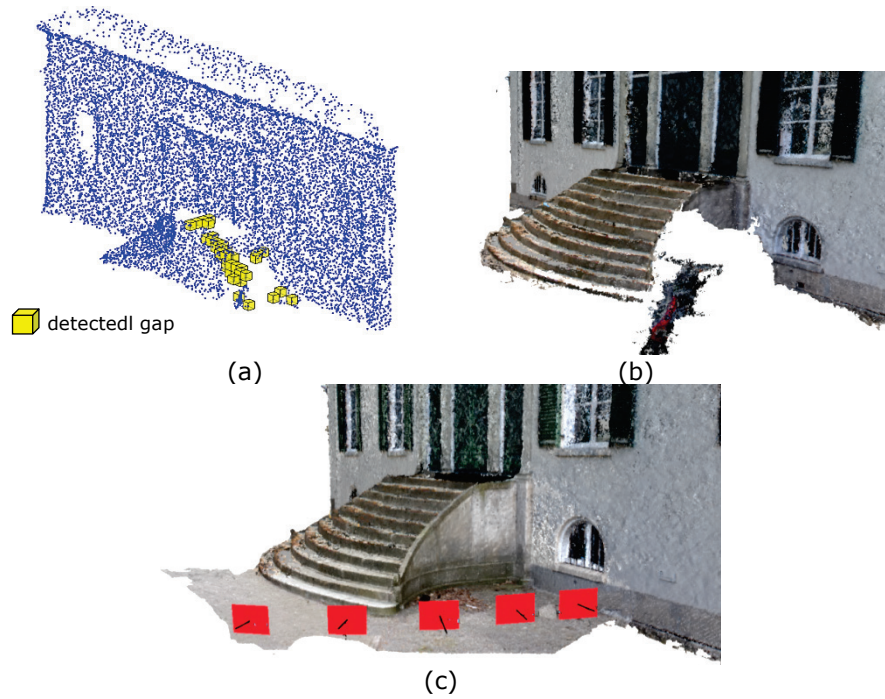


Fig. 6.28: The dense point cloud of façade. (a) Detected gap voxels. (b) Dense point cloud before detection (b) dense point cloud after the auxiliary image capture.

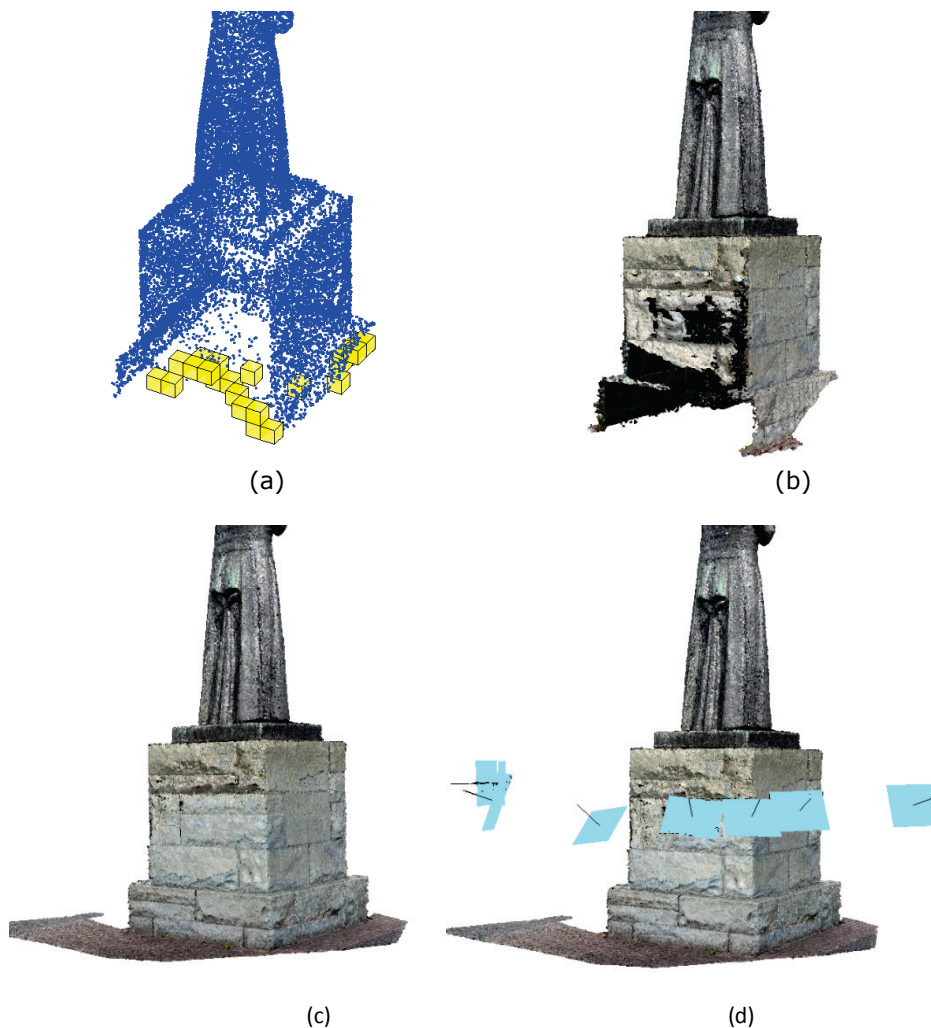


Fig. 6.29: The dense point cloud of the statue base. (a) The detected gap voxels. (b) The dense point cloud before recovery. (c) The dense point cloud after the auxiliary image capture shown in (d).

6.4 Summary

In this chapter, the detection of gaps in image-based point clouds were presented as well as a procedure to capture auxiliary images for the final 3D modelling. Two techniques were proposed, the first technique in section 6.2.1 was based on a 2D projection of the point cloud back into the images. Then, image processing techniques were used for detection. This technique was not sufficient to detect all the gap types in 3D space. The second method which represents the core method as discussed in section 6.2.2 was based on 3D space voxelization where the voxels are classified either as empty or occupied. Visibility analysis by testing the ray-voxel intersection were crucial

to discover the potential gap voxels. Different tests were presented for a façade in Fig. 6.21 and a statue in Fig.6.23. Point cloud with openings are also tested as shown in Fig.6.25 and Fig. 6.26. The voxel based technique for gap detection was not prepared to detect gaps in the poor textured areas. After the gap detection, a simple method was presented to capture auxiliary images for a final 3D modelling of gaps and missing parts.

Chapter 7 Case Study

This chapter describes and analyses a complete experiment in using the proposed methodology of this thesis in the image based modelling of a cultural heritage church. The church represents one of the oldest churches (approx. 1100AD) in the city centre of Enschede – Netherlands.

The approach used in this experiment was applied in a sequence of steps starting with a camera network design (chapter 4) that satisfied the necessary object coverage to produce a complete 3D model. This was followed by field guidance of the camera operator to capture the planned images in a flexible and simple way. The captured images were oriented automatically by using the bundle adjustment computations. The image orientation was based on finding sufficient image correspondences through the proposed guided matching techniques (chapter 5). The benefits of the guided image point correspondences compared to the full pairwise method were presented in terms of time consumption and accuracy of point localization. Image orientation was computed using the open source software *Apero* and the 3D point cloud of the church was created by using the techniques of dense matching with *Micmac* (Pierrot-Deseilligny 2012). Finally, gap detection was applied to the church point cloud and the final auxiliary images were captured. The processing was continued for the completion of the point cloud (chapter 6) and final 3D modelling.

The images were taken using a calibrated Canon EOS 500D with a resolution of 15 mega pixels (4752×3168 pixels). The illumination effect was reduced by capturing the images on a cloudy day with a 18 mm focal length. Details of the implementation sequence of the proposed 3D modelling are presented in the following subsections.

7.1 Referencing data

To verify the quality of the proposed method for 3D modelling, a benchmark was necessary to have an external validation for the accuracy and reliability of the produced models. Therefore, terrestrial laser scanning (TLS) was conducted around the church building using a “Trimble CX scanner” where the manufacturer single point positioning accuracy is 4.5 mm per 30 m distance.

The resulting point cloud consists of more than 23 million points. It has an average point spacing of 5mm and is shown in Fig. 7.1.

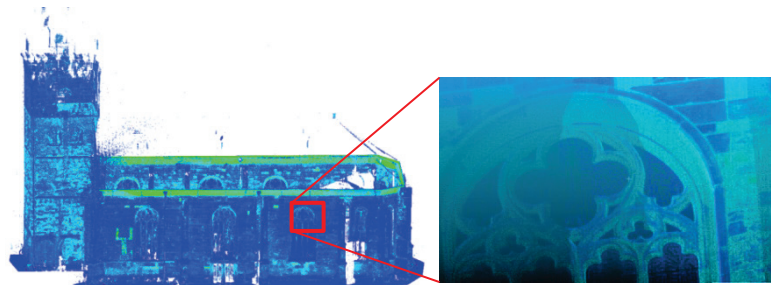


Fig.7.1: The reference point cloud by using the terrestrial laser scanning.

Six control points GCP with four checkpoints were also fixed on the church facades in a local coordinate system (Fig.7.2a). This is to ensure the correct referencing and scaling of the produced image based point cloud. The point heights were between 2-3 meters.

The control target point was designed as a 15 cm squared shape coloured with black and white (Fig.7.2b).

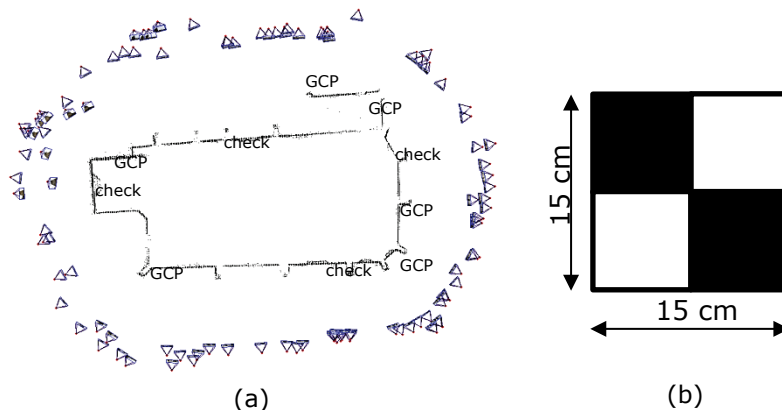


Fig.7.2: (a) The GCPs distribution. (b) The GCP target design.

7.2 Camera network design

The camera network design was started by using the empirical formula of equation (4.6). This was necessary to estimate the impact of the design on the final 3D model.

According to the demands of a highly detailed modelling and the accessible space around the church, the GSD was designed as 3-5 mm with a camera focal length of 18 mm. The expected accuracy in the object space was around 10 mm as shown in Fig.7.3. This is based on an assumed standard error of half a pixel in the measured SIFT points according to Barazzetti et al. (2010).

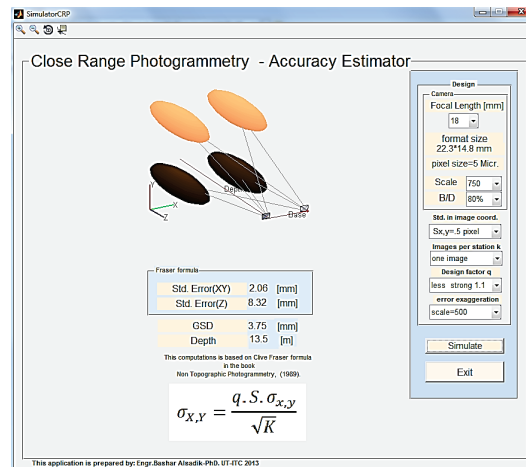


Fig.7.3: Accuracy estimation for the imaging design.

All computations are based on a rough point cloud derived from a video image stream (640×480 pixels) around the church using VSfM as shown in Fig. 7.4.

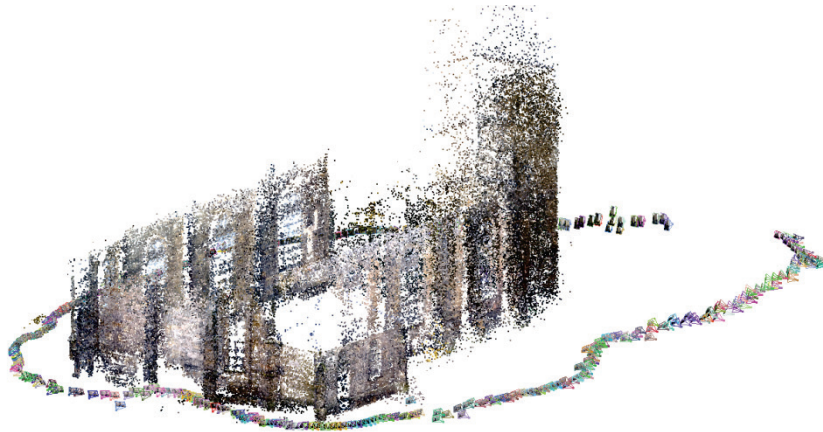


Fig.7.4: The rough point cloud of the church using VSfM.

The acquired rough point cloud was segmented into facades as shown in Fig. 7.5 to build the camera network. Some parts seem over-segmented or under-segmented and this will result in planning either more than necessary or less dense camera blocks respectively. However, the designed large number of cameras will guarantee the full coverage in both cases.



Fig.7.5: The segmentation of the church rough point cloud.

The designed network was based on the estimated accuracy to simulate a dense network (311 cameras) as shown in Fig.7.6a. The dense camera network was filtered according to the accuracy requirements and resulted in the imaging configuration of Fig. 7.6b, which consisted of 118 cameras. The optimization algorithm was applied to find the final imaging configuration as shown in Fig. 7.6c.

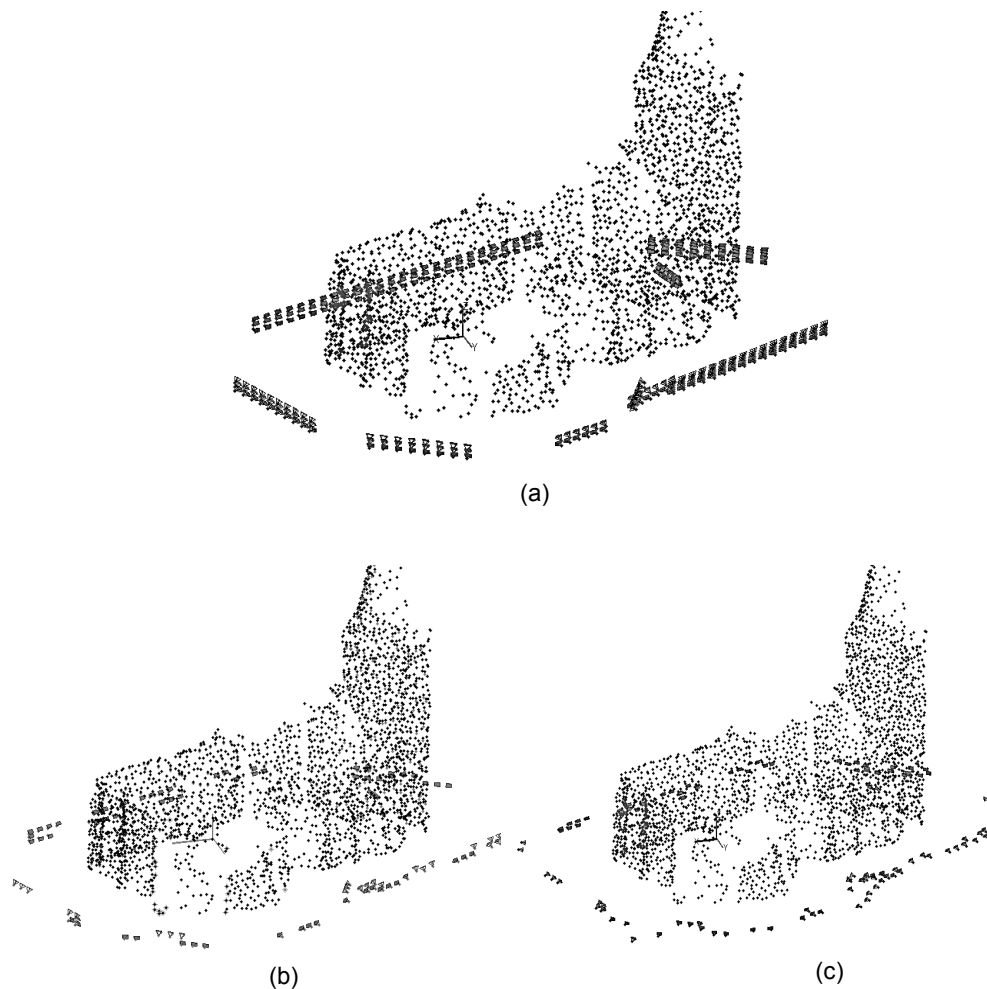


Fig.7.6: The camera planning and optimization of the church. (a) Dense network. (b) Filtered network. (c) Optimal network.

7.3 Guiding the image capture

Synthetic images were created as discussed in chapter 5 for every designed image as illustrated in the sample of Fig. 7.7a. These synthetic images were used as a guide to capture the real equivalent images (Fig. 7.7b).

Every new captured image was evaluated and the rejected images were recaptured. The procedure was continued until the complete set of images was taken. Fig. 7.7a,b illustrate a sample of both real and synthetic images matching. A space resection to compute the orientation of the real image for the validation is illustrated in Fig. 7.7c,d.

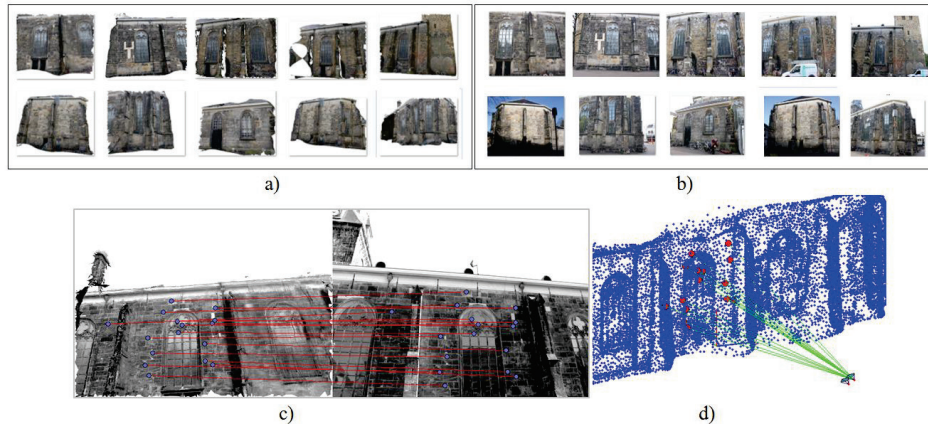


Fig.7.7: Sample of synthetic and real captured images. (a) Synthetic images. (b) Real images. (c) SIFT matching between the synthetic and captured image. (d) Space resection.

The complete set of the captured images was 118 images (Fig.7.8) which shows an acceptable (≥ 3 cameras) amount of coverage.

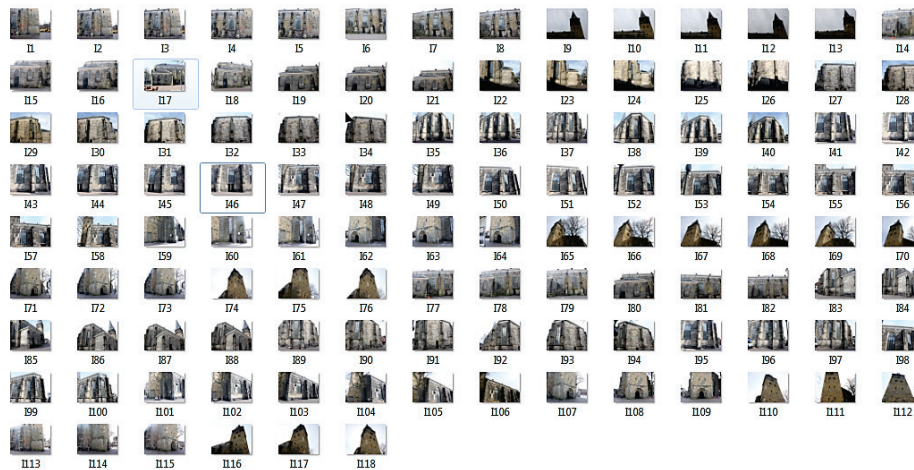


Fig.7.8: Captured data set for the 3D modelling of the church.

7.4 Extraction of image correspondences

The designed camera orientation and the 3D information about the church object are necessary to compute the image connectivity as mentioned in chapter 5. The computed connectivity in the case of full pairwise matching as well as the case of the guided matching is shown in Fig.7.9. This figure shows the computation reduction offered by the guided matching compared to the full pairwise where all the possible matches between the stereo pairs were tested.

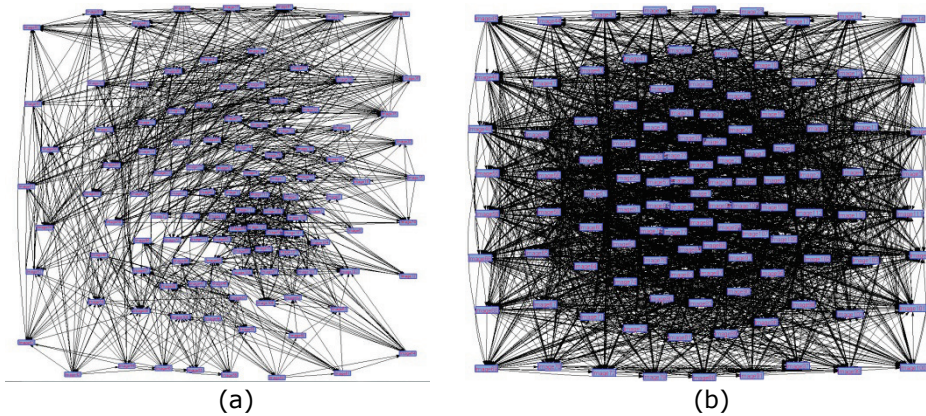


Fig.7.9: Comparison between full pairwise and guided connectivity graphs. (a) Connectivity graph of the Pre-designed network (1319 matches). (b) Full pairwise connectivity graph (6903 matches).

The graph in Fig.7.10 illustrates the time consumption in both cases of full pairwise and guided matching with three different image resolutions using the same computer. This graph shows that the time needed for the matching of the pre-designed camera network was about 70% less than the full pairwise matching.

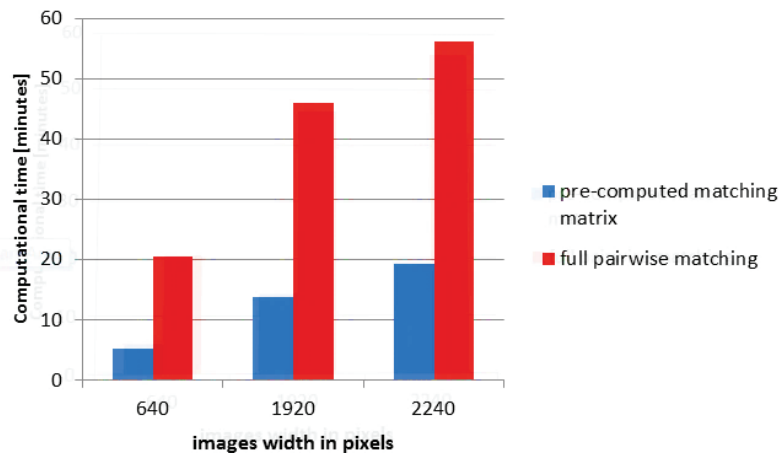


Fig.7.10: The time consumed in full pairwise and guided matching for the church dataset.

Moreover, the guided approach was advantageous to avoid possible mismatching as shown in Fig.7.11. This often happened in the church facades with its repetitive texture pattern (Fig. 7.11a). The symmetric connectivity matrix shown in Fig. 7.11b and Fig. 7.11c is another graphical representation of the image connectivity. These two figures show two cases of correct matching (green colour) and mismatching (red colour).

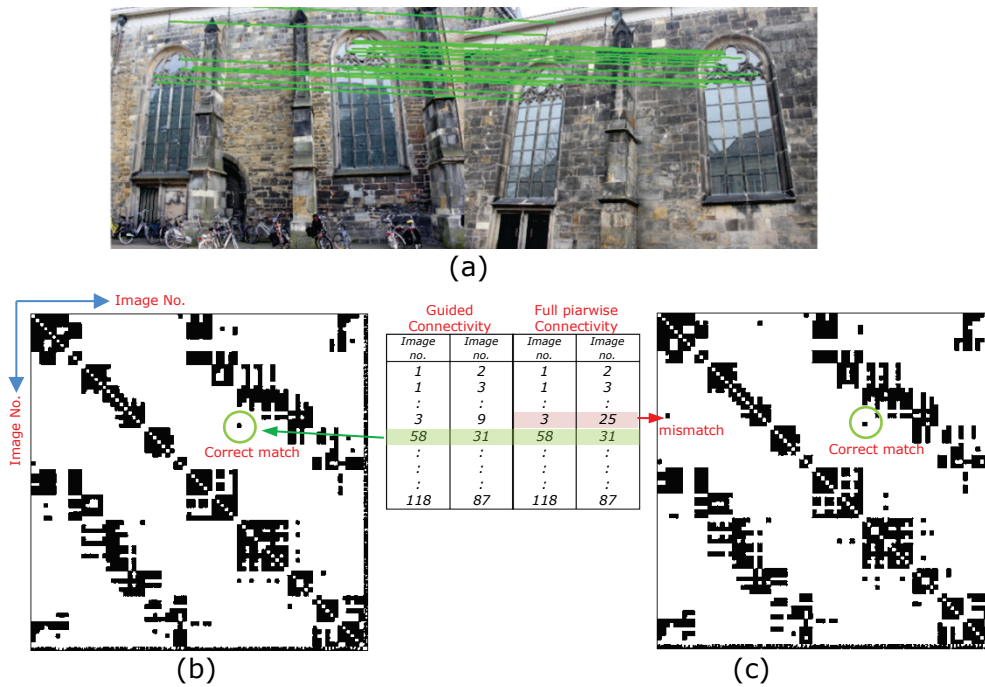


Fig.7.11: Mismatching case of a repetitive pattern of the church left and right facades. (a) SIFT mismatching in full pairwise. (b) Guided approach. (c) Full pairwise approach.

Further benefits were attained from the guided matching besides the time consumption and the mismatching avoidance. These benefits were mainly in the higher correctness within the matching of the image correspondences and the high number of the detected matching points.

A window size of 150 pixels was applied to compute the correspondences between the short baseline images. For wide baseline images as in Fig. 7.12a, a window size of 250 pixels was applied and resulted in 14882 point correspondences. This is a large number of homologous points compared to the points acquired by other software. Fig. 7.12b shows 722 matching points result with the same stereopair using SIFT with open source VSfM. Fig. 7.12c shows 2543 matching points with the same stereopair using Agisoft photoscan software.



Fig. 7.12: (a) Image correspondences using our guided matching. (b) Image correspondences using VSFM software. (c) Image correspondences using Agisoft photostan software.

7.5 Image orientation

The number of the image correspondences computed using the guided matching in the experiment was high and therefore reduced to less than half (≈ 200000 points) to have a reasonable processing data and time. The open source software APERO was used for the image orientation which is a part of the full package of MICMAC (Pierrot-Deseilligny 2012). APERO was fed with the image correspondences according to the guided approach described in chapter 5. A relative orientation was first applied and then followed by the absolute orientation using 6 GCP and keeping the other four as checkpoints (Fig. 7.13).

The GCP target points were marked manually on the corresponding images which was probably a source of pointing error. However, careful zooming and marking was applied on the images to decrease the chance of large errors.

After feeding the data to APERO software, the bundle adjustment was applied to obtain the final oriented camera network and the sparse point cloud as shown in Fig.7.13. The unit standard deviation of the adjustment was 0.82 pixels which is higher than the assumed standard deviation of 0.5 pixels in the designing step. One reason could be the insufficient calibration parameters of the camera. Therefore, an assumed measurement precision of 1 pixel in the image coordinates seems more reasonable. The predicted precision in the object space equals 2.5 mm according to the mean image scale. This amount met the designing criteria and the selected GSD in section 7.1.

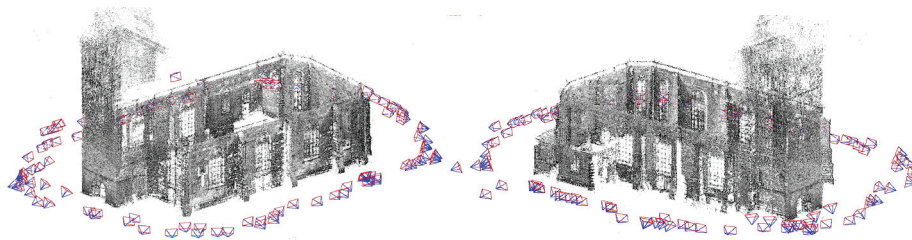


Fig.7.13: The oriented camera network by guided image correspondences using APERO.

To investigate the predicted accuracy, the root mean square error RMSE was computed for the GCPs and checkpoints as illustrated in Table 7.1.

Table 7.1: RMSE of the GCPs and checkpoints.

Point	D_x [m]	D_y [m]	D_z [m]	Point error
GCP 12	0.001	-0.007	-0.005	0.009
GCP 20	0.001	-0.006	0.002	-0.006
GCP 24	0.003	-0.001	-0.000	0.003
GCP 31	-0.003	0.007	0.005	0.009
GCP 40	0.005	-0.009	0.009	0.014
GCP 53	0.001	0.004	0.000	0.004
	0.004	0.006	0.004	RMSE=0.008
checkP16	-0.011	-0.005	-0.008	0.015
checkP25	0.006	0.001	0.003	0.007
checkP51	0.002	-0.001	-0.002	0.004
checkP66	0.023	-0.001	0.011	0.026
	0.013	0.003	0.008	RMSE=0.016

The final 3D dense point cloud was generated by using the open source software MICMAC as shown in Fig. 7.14.

However, MICMAC cannot automatically produce the total point cloud directly after image orientation. Therefore, sub clouds were created first based on the image connectivity and then the total point cloud was prepared by manual fusing of the sub point clouds.

This manual fusing is the reason behind the difference of texture above the church entrance. Another reason is the illumination change during the capture of images of that part. The density of points was very high and sampled to finally have about 40 million points.



Fig.7.14: The dense textured point cloud of the church using MICMAC.

To validate the acquired point cloud, the distances between sub point clouds and their TLS reference point cloud were computed using the CloudCompare software. This comparison is based on using the Iterative Closest Point (ICP) algorithm to register the image based point cloud to the reference point cloud from TLS. Then, each point in the image based cloud is assigned to its nearest neighbour in the reference cloud and the distance metric is computed.

Two sub clouds were selected for the validation. The first point cloud of a sundial (Fig.7.15) shows a mean shift distance of 7mm with a standard deviation of 5 mm between the image based cloud (296284 points) and the TLS point cloud (559458 points).

The second test (Fig.7.16) of a church window indicates a similar accuracy of (8 ± 5) mm between the image based cloud (1987529 points) and the TLS cloud (3218109 points). These two tests indicate a highly accurate point cloud of the church building when compared to the reference TLS point cloud. The dominant blue colour in Fig. 7.15c and Fig. 7.16c visually indicates that

the image based point cloud are coinciding with the TLS point cloud. The amount of shift in both test agrees with the RMSE in the GCPs and the designed accuracy of the project of 10 mm in section 7.2.

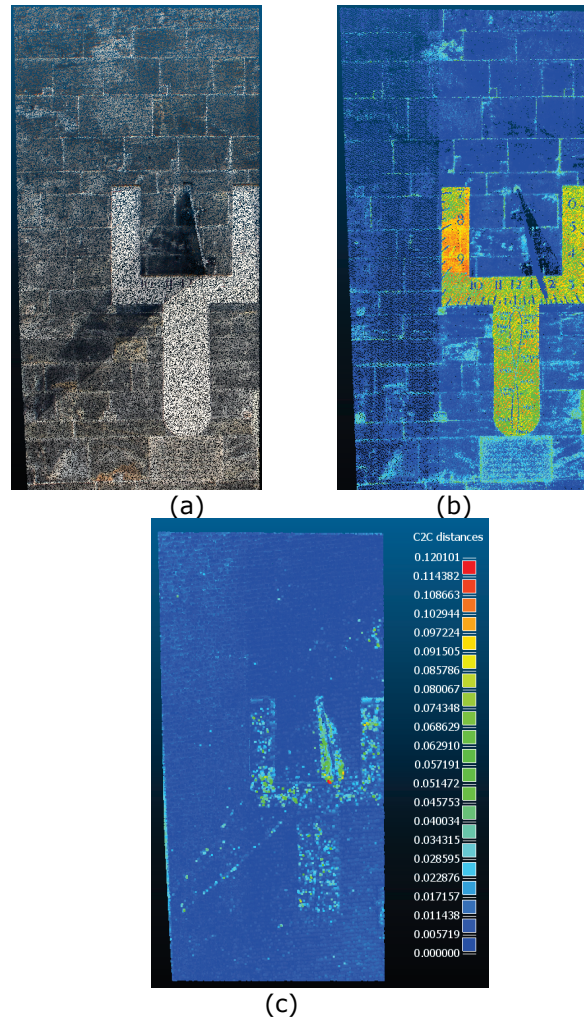


Fig. 7.15: The first test for point cloud validation. (a) Image-based point cloud. (b) TLS point cloud. (c) Distance between the point clouds.

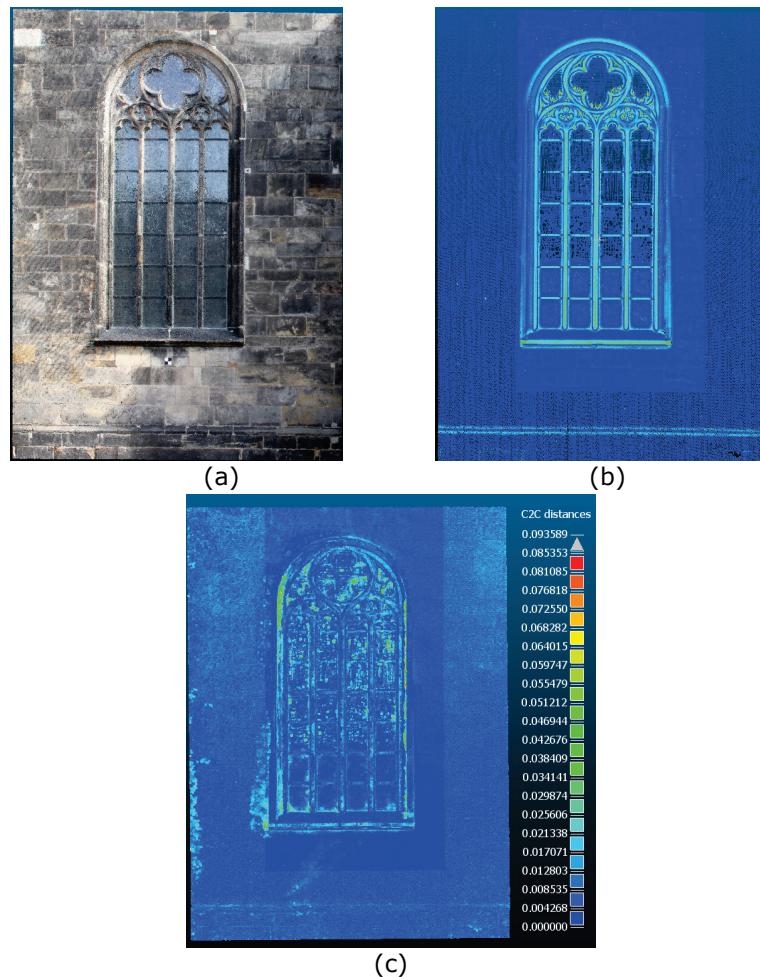


Fig. 7.16: The second test for point cloud validation. (a) Image-based point cloud. (b) TLS point cloud. (c) Distance between the point clouds.

To evaluate the point cloud created in the upper parts of the church, the following sub-cloud was selected as shown in Fig. 7.17.

The computed distance between the image based cloud (104220 points) and the TLS cloud (265856 points) is 3 ± 2 cm. This result indicates a poor accuracy of the created point cloud at the church tower when compared to the reference TLS point cloud. Moreover, the density of the points in the upper parts of the building is less than the density in the lower parts. This is also clear from the mentioned number of points in the three tests. This is expected because the images were captured at the street level and then the created point cloud affected by the weak angle intersection geometry and the reduced resolution at these upper parts.

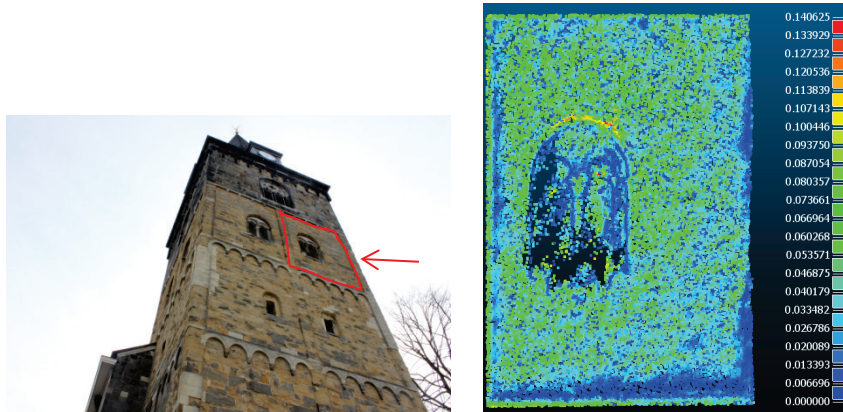


Fig. 7.17: The validation of the upper part of the church point cloud.

7.6 Gap detection and final modelling

The church architectural design as shown in Fig. 7.14 was characterized by the extruded columns on the facades of the church. These extruded columns represented a challenge for the 3D modelling because of the difficulty to accommodate them during the camera planning. This insufficiency was related to the use of a rough point cloud during the design stage.

Therefore, the few missing gaps in the dense point cloud especially near the column sides were first detected by the voxelization procedure (chapter 6) and then auxiliary images were taken. The gap detection was applied per facade to reduce the computation time.

Figure 7.18a illustrates the point cloud of one façade viewed by 13 images and was tested for gaps. Obviously, the façade point cloud included only gaps caused by occlusions and insufficient camera network coverage. The point cloud was voxelized and then the voxels were labelled according to the visibility as either empty voxels in red, partly occluded in yellow or occupied in green as shown in Figure 7.18b respectively. The filtering strategy was based on the visibility, neighbourhood index and altitude resulted in the final empty voxels which represented the occluded gaps in the point cloud (Fig. 7.18c).

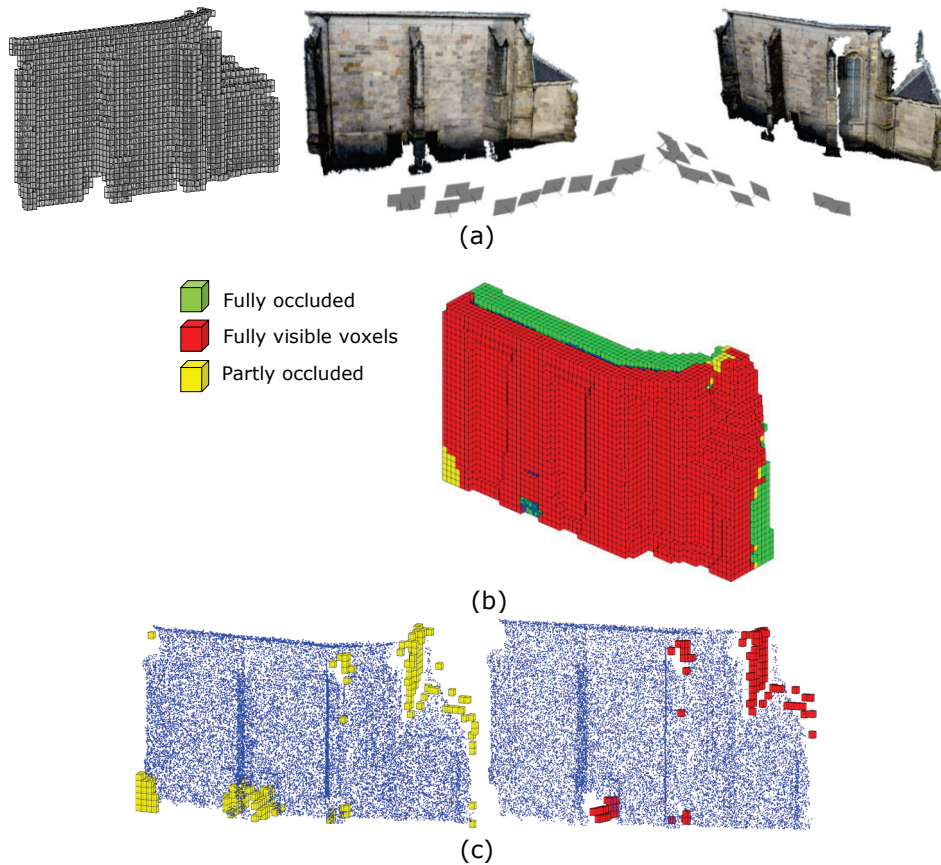


Figure 7.18: Gap detection with voxels. (a) The dense point cloud and occupied voxels. (b) The empty voxels are coloured according to visibility test. (c) Filtered voxels after visibility test (yellow) and the final detected gaps (red).

The total detected gaps represented as voxels were projected to the dense point cloud of the church as shown in Fig.7.19. It must be noted that the smallest part in the front façade in the second view of Fig. 7.19 is not represented as gap voxels. The gaps were neglected because this part was occluded by a tree which cannot be avoided.

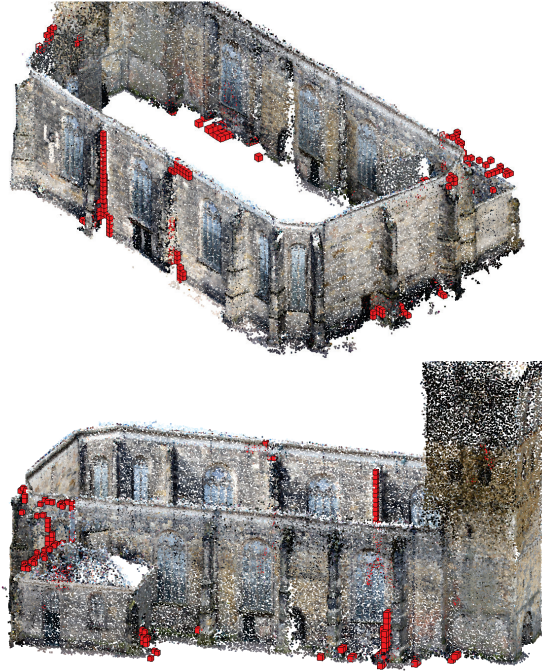


Fig. 7.19: The church point cloud with potential detected gaps represented as red voxels.

After the gap detection, auxiliary images were planned to be taken in a similar procedure described in chapter 6. The connectivity graph was prepared between the images involved with every detected gap location and the intended auxiliary images. The number of the auxiliary images was 63 added to the images planned before to have a total set of 175 images as shown in Fig. 7.20.

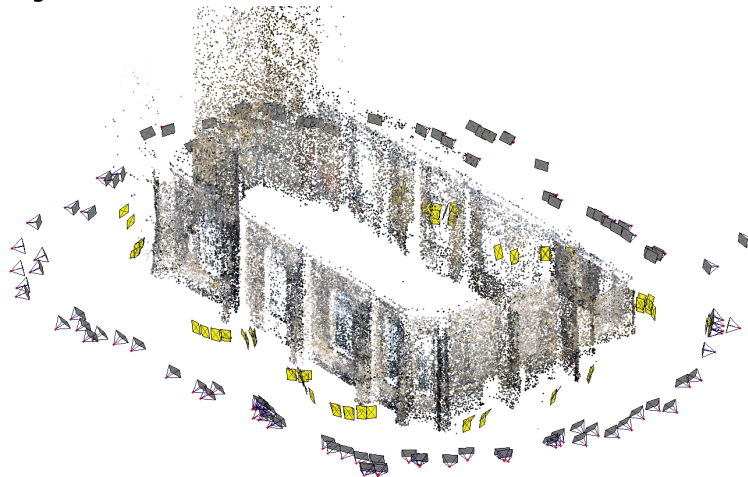


Fig. 7.20: The captured auxiliary images in yellow.

Fig. 7.21, to Fig. 7.23 illustrated a comparison between the dense point cloud before and after the auxiliary image capture at the detected gap areas.



(a)



(b)

Fig. 7.21: Dense point cloud illustrates the first detected gap in a column. (a) Before gap detection. (b) After adding the auxiliary images and modelling.



(a)



(b)

Fig. 7.22: Dense point cloud after the second gap detection. (a) Before gap detection and point cloud completeness. (b) After adding the auxiliary images (red) and modelling.



Fig. 7.23: Dense point cloud illustrates the gap in different parts of the church before (left) and after (right) gap detection and point cloud completeness.

The final point cloud after gap detection and auxiliary image capturing is illustrated in Fig. 7.24. The point cloud of the facades within the accessible camera views were completed. However, clusters of points were missing on the higher parts, especially on the tower clock. This was unavoidable because of the limited camera altitude from a street level as stated earlier in chapter 1. Moreover, some small parts near the ground were also missing because of the occlusion caused by people parking their bicycles very close to the walls. This problem can be avoided for real cultural heritage projects.



Fig.7.24: The complete point cloud of the church from a layman camera height.

7.7 Summary

The proposed method of this thesis was tested on a church in Enschede. The results show the sufficiency of this method for 3D image based modelling in terms of completeness and accuracy. For accuracy validation, internal and external tests were conducted. The internal accuracy test was applied by checking the RMSE for the GCPs and four checkpoints (Table 7.1). The RMSE was 8 mm for GCPs and 16 mm for checkpoints respectively. However, the RMSE of 8mm was computed if the largest shift of checkpoint 66 excluded. Furthermore, the external validation was conducted on the produced dense point cloud compared to a highly accurate reference point cloud produced by laser scanning. Three sample point clouds were selected as illustrated in Fig. 7.16, Fig.7.17 and Fig. 7.18. The mean shift distances in the first two tests were 7 mm and 8 mm respectively. However, for the upper parts of the church tower, the accuracy was deteriorated to 3 cm.

To complete the model, gap detection was applied as discussed in chapter 6 for the church point cloud. There were few gap areas detected in the produced point cloud. Auxiliary images were taken for every detected gap and an improved coverage was attained in the final point cloud as illustrated in Fig. 7.21 to Fig. 7.24.

Chapter 8. Conclusions and recommendations

This chapter presents the conclusions to the objectives and research questions presented in chapter 1 based on the research findings of chapters 4-7.

8.1 Conclusions

An imaging system with an automated camera network design method is presented in this thesis. This multi imaging system is developed for the 3D modelling of objects like buildings and statues. Accordingly, cultural heritage documentation and preservation is a major application area of such a technique. The aim was also to enable non professionals to conduct this task. The implementation of the proposed method as discussed in the thesis chapters resulted in the following conclusions:

- The proposed method proved its efficiency as discussed in the thesis chapter 4 through building the optimal camera network. The optimality meant to find the minimum number of cameras with the required accuracy in the object points. The practicality of capturing the designed images was also proposed as discussed in chapter 5 to enable even non-professionals to successfully create 3D models. The completeness of the predicted 3D model was studied in chapter 6 by detecting gaps in the point cloud and adding auxiliary images to finalize the necessary photo capture.
- For the camera network design step, minimal network computations are proposed by using either the requirements of accuracy in the object space or the higher coverage requirements. Both methods gave efficient results, but with a preference for the method of filtering for accuracy.
- The optimization technique proved the efficiency to converge to a higher accuracy criteria if needed. However, careful selection of the camera bounds was crucial in convergence. The user is able to select the suitable proposed constraints presented in chapter 4 based on the project needs.
- The methodology was based on using a good pre-calibrated camera. This is expected to improve the accuracy of the image orientations and the produced models.
- The proposed segmentation method in chapter 4 was based on a 2D processing of the point cloud. This was not efficient in segmenting all the

types of buildings, especially when the roofs are represented in the point cloud.

- The automated camera network design is suitable for autonomous mapping vehicles like UAVs where the flight time is limited. When, UAVs are equipped with a GNSS and an IMU, it can be programmed to realize the camera locations and angular orientations of the designed optimal network. In this case the guidance by the synthetic images is not needed. Accordingly, producing a highly efficient and complete 3D model out of these UAV-based images is expected to improve.
- Synthesizing the designed optimal images was very useful for the fieldwork to capture their equivalent real images. Some problems were found when the SIFT matching between the synthetic images and the real images was not successful. Visual inspection was the easiest way to overcome such a problem.
- The multi window guided SIFT matching produced promising results for the image orientation. Some challenging cases with wide baseline imaging have been processed with this technique. Higher number of tie points were offered when compared with the conventional technique beside reducing the processing time of tie points matching. Moreover, some of the state-of-the-art packages were not able to orient the imaging network of the church object of chapter 7 while our guided approach succeeded. The failure in these packages was either because of the repetitive pattern effect or the wide baseline configuration which is solved in our approach.
- The automated gap detection technique presented in this thesis was efficient to find the incomplete parts of the modelled object. Some difficulties were faced when there were real openings or poorly textured parts of the object. The detected gaps offered a useful guidance for a subsequent image capture to complete the missing parts in the 3D model.
- The research was intended for heights between 0.5 to 2 meters that are accessible by a layman. This limitation affected the produced 3D models, especially in the upper parts and roofs of buildings. Therefore, the produced models of building were mainly for their facades. However, this limitation can be lifted when using a UAV which offers an accessible fly above the buildings and higher parts of objects.
- The proposed gap detection technique based on a 2D image processing was less efficient compared to the 3D voxel based technique. However,

the voxelization was expensive in terms of computational time and computer memory.

- The effect of occlusions on the completeness of the 3D model and their avoidance was not considered.

8.2 Recommendations

Based on the discussion and experiments of the proposed imaging technique, we can recommend to:

- Use other sources like available CAD models in representing the object to design the camera network. This is appropriate in some cases when the video based rough model is not applicable.
- Consider more flexibility regarding accessible heights for the camera like when using UAVs. This will be more efficient to cover the upper parts and roofs of objects.
- Investigate the application of camera self-calibration during the camera network design and the subsequent image orientation.
- Apply the point cloud segmentation in three dimensions rather than two dimensions for a better camera network design that can accommodate for roofs as well.
- Add a tool of gap detection in the state of the art packages of the image based modelling. This is by graphically representing the detected gaps by voxels as shown in chapters 6 and 7. Such a tool will be very useful for users to complete their 3D models, especially on sites with limited access, like archaeological sites and museums.
- Improve the camera network design by considering the avoidance of occluding objects. This will improve the completeness of the produced 3D models.
- Design the proposed imaging system in a real time implementation. This is a matter of advanced hardware and software abilities. Moreover, programming our method as a smartphone/tablet app will be more suitable for the actual fieldwork.
- Use the Least Square Matching (LSM) method in the guided multi window matching to refine the localization of the matched SIFT points to sub-

pixel accuracy. This is expected to improve the image orientation accuracy as proposed in (Barazzetti et al. 2010) .

- Support the guided imaging with the necessary instructions that can be printed on a screen to give some extra information like the needed distance between the object and the camera and/or the 2D imaging network plan as shown in the sketch of Fig. 8.1.

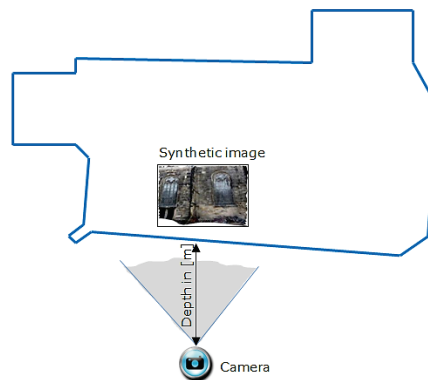


Fig 8.1: 2D plan illustration to capture images with the suggested guidance.

This thesis showed that the proposed imaging system is suitable for the applications of highly detailed 3D image based modelling in terms of automation and practicality. State-of-the-art software can add the developed functionalities of camera network design, guidance and gap detection to their current components. The proposed interactivity between the camera operator and the software will certainly produce the required efficiency of 3D modelling.

Appendix A

A1- Optimization

Optimization techniques are used to find a set of design parameters, $x = (x_1, x_2, \dots, x_n)$, that can be defined as optimal. In a simple case this might be minimization or maximization of some system characteristic that is dependent on x . In a more advanced formulation the objective function $f(x)$ to be minimized or maximized, might be subject to constraints in the form of equality constraints, $h_i(x) = 0$ ($i = 1, \dots, m_e$), inequality constraints, $g_i(x) \leq 0$ ($i = m_e + 1, \dots, m$); and /or parameter bounds, x_i, x_u .

The problem that is difficult to solve is the Nonlinear Programming (NP) problem in which the objective function and constraints can be nonlinear functions of the design variables as in the case of the camera placement problem.

The solution of nonlinear unconstrained minimization problem or nonlinear least square problem in photogrammetry and computer vision is common and crucial when solving the bundle adjustment problem in the orientation computations and SfM. This is either to be solved by *Levenberg – Marquardt* LM or by *Gauss – Newton* methods (K. Madsen et al. 2004). However, the constrained minimization is harder to solve from a mathematical viewpoint because of constraints that have to be considered in the solution. This constrained minimization problem is to be solved by introducing the LaGrange multipliers as follows:

$$L(x, \lambda) = f(x) + \sum \lambda_{g,i} g_i(x) + \sum \lambda_{h,i} h_i(x) \quad (A1)$$

The vector λ , which is the concatenation of λ_g and λ_h , is the LaGrange multiplier vector and its length equals the total number of constraints

It is well known in algebra that the gradient must be zero at a minimum, however, for a complicated minimization analogous conditions called Karush-Kuhn-Tucker (KKT) conditions must be fulfilled to find the optimum solution for this problem and guarantee a global optimum (Rao 2009). The KKT conditions are stated that:

$$\left. \begin{array}{l} \nabla_x L(x, \lambda) = 0 \\ \lambda_{g,i} g_i(x) = 0 \\ g(x) \leq 0 \\ h(x) = 0 \\ \lambda_{g,i} \geq 0 \end{array} \right\} \quad (A2)$$

It must be noted that LaGrange multipliers λ turn the inequality function into equality because it defines a stationary point where the partial derivatives are zero. Thus, yields a necessary condition for optimality in constrained problems.

The mathematical challenge to solve such kind of problems is mainly in the existence of inequality nonlinear constraints as in our case of camera network optimization. Different methods are currently used to solve the constrained optimization problems like: trust-region, sequential quadratic programming (SQP) and interior point techniques (Waltz et al. 2006; Rao 2009).

According to the literature, the approach of interior point has enjoyed great success and proved to be effective for a variety of problem classes due to their regularization effects to the constraints. Interior-point methods according to Curtis (2011) have become the workhorse in large-scale optimization due to their Newton-like qualities in terms of scalability and convergence performance.

This interior point algorithm is to solve a sequence of approximate minimization problems and essentially implemented by using a barrier method as follows (Byrd et al. 1999):

$$\begin{aligned} & \underset{x,s}{\text{minimize}} f(x) - \mu \sum_{i=1}^m \ln(s_i) \\ & \text{Subject to:} \\ & h(x) = 0 \\ & g(x) + s = 0 \end{aligned} \tag{A3}$$

Where μ is the barrier parameter and where the slack variables s are assumed to be positive to keep $\ln(s_i)$ bounded. As μ decreases to zero, the minimum of f_μ should approach the minimum of f . The inequality constraints are approximated to equality constraints by the addition of the slack variables to make it easier to solve. To update the unknown parameters (camera exterior orientation x) and the slack variables s , a step length must be determined and added to the initial values of these parameters ($x + d_x, s + d_s$) in an iterative way. These steps can be solved either by a direct (linear) search or by a conjugate gradient CG technique. At each iteration, the algorithm decreases a merit function to determine whether the total step is acceptable as shown:

$$f_\mu(x, s) + v \|h(x), g(x) + s\| \tag{A4}$$

Where v is called merit or penalty parameter. The solution is accepted by checking if the attempted step doesn't decrease the merit function, and in

that case the algorithm rejects the step and attempt a new step until the stopping tolerance satisfied.

It's worth to mention that we used the optimization toolbox of Matlab to implement this technique through the use of the function (fmincon) (Matlab 2010).

A2- Cost function in optimal camera network design

In this section, we will discuss the formulation of the objective function of the optimal camera network.

As mentioned in chapter 4, the mathematical model is represented by collinearity equations, and can be linearized by the first-order development of Taylor series. Moreover, with the redundant observations, least square method is usually followed to compute the adjusted parameters. This will lead to the following observation equation model (Ghilani and Wolf 2006)

$$Av + B\Delta = f \quad (A5)$$

Where

A : The matrix of partial derivatives of collinearity with respect to the observations.

B : The matrix of partial derivatives of collinearity with respect to the unknowns.

v : The vector of residual errors.

Δ : The vector of corrections.

f : The vector of function evaluations.

Accordingly, in this thesis, the objective function is formulated by simplifying the normal equation matrix in least squares L.S. adjustment method into two groups of parameters. Namely, Δ^e, Δ^s represent the corrections to the exterior orientations 'e' and the object coordinates 's' respectively. This yields the following weighted L.S. normal equation.

$$\begin{bmatrix} [B_e^T W B_e] & [B_e^T W B_s] \\ [B_s^T W B_e] & [B_s^T W B_s] \end{bmatrix} = \begin{bmatrix} Q_e & Q_{es} \\ Q_{se} & Q_s \end{bmatrix} \quad (A6)$$

Where W is the matrix of weights. The designed optimization objective of increasing the precision is to achieve an optimal form of Q_s which is the covariance matrix of object point coordinates (X, Y, Z) . Consequently, this will guarantee the maximum accuracy (Fraser 1989) with the optimal minimum number of cameras. This objective function is expressed in equation 4.7 $f(x) = \min(\sum_{i=1}^n Q_s^i / n)$ by minimizing the computed average error in $(X, Y, \text{ and } Z)$ within n unknown object points, which is computed according to the collinearity equations model.

Bibliography

- Acute3D (2013). Smart3DCapture. <http://www.acute3d.com/software/>.
- Agarwal, S., Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz and R. Szeliski (2011). "Building Rome in a day." Communications of the ACM 54 (10): 105-112.
- Amini A. Sh., S. M., Varshosaz M. (2010). Optimum Selection of Images in Industrial Photogrammetry Using Fuzzy Computation. Saint-Mandé, France. XXXVIII.
- ARC3D (2006). Automatic Reconstruction Cloud., KU Leuven. <http://www.arc3d.be/>
- Autodesk (2012). 123D Catch <http://www.123dapp.com/catch>.
- Autodesk (2013). RECAP. <http://www.autodesk.com/products/recap/overview>.
- Barazzetti, L., M. Scaioni and F. Remondino (2010). "Orientation and 3D modelling from markerless terrestrial images: combining accuracy with automation." The Photogrammetric Record 25(132): 356-381.
- Bartelsen, J., H. Mayer, H. Hirschmüller, A. Kuhn and M. Micheline (2012). "Orientation and Dense Reconstruction of Unordered Terrestrial and Aerial Wide Baseline Image Sets." ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. I-3: 25-30.
- Bay, H., A. Ess, T. Tuytelaars and L. Van Gool (2008). "Speeded-Up Robust Features (SURF)." Computer Vision and Image Understanding 110(3): 346-359.
- Bernardini, F., J. Mittleman, H. Rushmeier, u. Silva and G. Taubin (1999). "The Ball-Pivoting Algorithm for Surface Reconstruction." IEEE Transactions on Visualization and Computer Graphics 5(4): 349-359.
- Bittner, J. and P. Wonka (2003). "Visibility in computer graphics." Environment and Planning B: Planning and Design 30(5): 729-755.(Cohen-Or et al. 2003)
- Byrd, R. H., M. E. Hribar and J. Nocedal (1999). "An Interior Point Algorithm for Large-Scale Nonlinear Programming." SIAM J. On Optimization 9 (4): 877-900.
- Chandler, J. and J. Fryer. (2011). "Accuracy of AutoDesk 123D Catch." Retrieved 20-03-2012, 2012.
- CIPA. (2011). "The International Committee for Documentation of Cultural Heritage ", from <http://cipa.icomos.org/>.
- Cohen-Or, D., Y. L. Chrysanthou, C. T. Silva and F. Durand (2003). "A survey of visibility for walkthrough applications." IEEE Transactions on Visualization and Computer Graphics 9(3): 412-431.
- Cohen, A., C. Zach, S. N. Sinha and M. Pollefeys1 (2012). Discovering and exploiting 3D symmetries in structure from motion. Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society: 1514-1521.

- Curtis, F. E. (2011). "A Penalty-Interior-Point Algorithm for Nonlinear Constrained Optimization." *Mathematical Programming Computation*.
- Dellaert, F., S. Seitz, C. E. Thorpe and S. Thrun (2000). Structure from Motion without Correspondence. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'00)*.
- Dunn, E., G. Olague and E. Lutton (2006). "Parisian camera placement for vision metrology." *Pattern Recognition Letters* 27(11): 1209-1219.
- EOSsystems (1994). PhotoModeler www.photomodeler.com.
- Fisher, P. F. (1996). "Extending the Applicability of Viewsheds in Landscape Planning" *Photogrammetric Engineering and Remote Sensing (PE&RS)* 62(11): 1297-1302.
- Forstner, W. (2002). Computer vision and photogrammetry, Mutual questions, Geometry, statistics and cognition. *International Symposium Photogrammetry meets Geoinformatics*.
- Fraser, C. S. (1982). "Optimization of Precision in Close-Range Photogrammetry." *Photogrammetric Engineering and Remote Sensing* 48 (4): 561-570.
- Fraser, C. S. (1984). "Network Design Considerations for Non-Topographic Photogrammetry." *Photogrammetric Engineering and Remote Sensing* 50 (8): 1115-1126.
- Fraser, C. S. (1989). *Non Topographic Photogrammetry*, Edwards Brothers Inc.
- Fryer, J., et al. (2007). *Applications of 3D Measurement from Images*. Scotland,UK, Whittles Publishing.
- Furukawa, Y. and J. Ponce (2009). "Accurate, Dense, and Robust Multiview Stereopsis." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(8): 1362-1376.
- Furukawa, Y. and J. Ponce (2010). PMVS, <http://grail.cs.washington.edu/software/pmvs/>.
- Ghilani, C. D. and P. R. Wolf (2006). *Adjustment Computations: Spatial Data Analysis*, John Wiley & Sons Inc.
- Ghuneim, A. G. (2010). "Contour Tracing." Retrieved 10-12 2011, from http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/contour_tracing_Abeer_George_Ghuneim/index.html.
- Gilboa, A., A. Tal, I. Shimshoni and M. Kolomenkin (2013). "Computer-based, automatic recording and illustration of complex archaeological artifacts." *Journal of Archaeological Science* 40(2): 1329-1339.
- Guidi, G., M. Russo and D. Angheluddu (2014). "3D survey and virtual reconstruction of archeological sites." *Digital Applications in Archaeology and Cultural Heritage* 1(2): 55-69.
- Haala, N. (2011). *Multiray Photogrammetry and Dense Image Matching*. Photogrammetric Week 2011. Wichmann Verlag, Berlin/Offenbach: 185-195.

- Hammoudi, K., F. Dornaika, B. Soheilian, B. Vallet, J. McDonald and N. Papanicolaou (2012). "Recovering Occlusion-Free Textured 3D Maps of Urban Facades by a Synergistic Use of Terrestrial Images, 3D Point Clouds and Area-Based Information." *Procedia Engineering* 41(0): 971-980.
- Hartley, R. and A. Zisserman (2004). *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN: 0521540518.
- Hirschmuller, H. (2005). *Accurate and efficient stereo processing by semi-global matching and mutual information*. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*.
- Hirschmuller, H. (2008). *Stereo Vision in Structured Environment by Consistent Semiglobal Matching*. *Proceeding of the IEEE Conference on Computer vision and pattern recognition CVPR*. New York. 2: 2386-2393.
- Hullo, J. F., P. Grussenmeyer and S. Fares (2009). *Photogrammetry and Dense Stereo Matching Approach Applied to The Documentation of The Cultural Heritage Site of Kilwa (Saudi Arabia)*. XXII CIPA Symposium, Kyoto, Japan, ISPRS.
- Huq, S., A. Koschan and M. Abidi (2013). "Occlusion filling in stereo: Theory and experiments." *Computer Vision and Image Understanding* 117(6): 688-704.
- Impoco, G., P. Cignoni and R. Scopigno (2004). *Closing gaps by clustering unseen directions*. *Shape Modeling Applications, 2004. Proceedings*.
- Joy, K. I. (1999). *The Depth-Buffer Visible Surface Algorithm*. *Computer Science Department, University of California*. : 1-5.
- Katz, S., A. Tal and R. Basri (2007). "Direct Visibility of Point Sets." *ACM Transactions on Graphics* 26(3).
- Kazhdan, M., M. Bolitho and H. Hoppe (2006). *Poisson surface reconstruction*. *Proceedings of the fourth Eurographics symposium on Geometry processing*. Cagliari, Sardinia, Italy, Eurographics Association: 61-70.
- Kiamehr R. (2003). *Multi Object Optimization of Geodetic Network*. *Proc. NCC Geomatics 82 conferences*. Tehran, Iran.
- Kosecka, J. and W. Zhang (2010). "Extraction, matching, and pose recovery based on dominant rectangular structures." *Computer Vision and Image Understanding* 100: 274-293.
- Kuzu, Y. (2004). *Volumetric Object Reconstruction by Means of Photogrammetry*. PhD thesis, Technical University of Berlin.
- Lee, B.-G., Liliana and D.-H. Shin (2010). "Enhanced computational integral imaging system for partially occluded 3D objects using occlusion removal techniques and recursive PCA reconstruction." *Optics Communications* 283 (10): 2084-2091.

- Letellier, R. (2007). Recording, Documentation, and Information Management for the Conservation of Heritage Places, Guiding Principles, Library of Congress Cataloging-in-Publication Data.
- Lhuillier, M. and L. Quan (2002). "Robust Dense Matching using Local and Global Geometric Constraints." *Pattern Analysis and Machine Intelligence, IEEE_24* (8): 1140 - 1146.
- Lourakis, M. I. A. and A. A. Argyros. (2004). "The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg - Marquardt Algorithm. ."
- Lowe, D. G. (1999). Object Recognition from Local Scale-Invariant Features. *Proceedings of the International Conference on Computer Vision- Volume 2 - Volume 2, IEEE Computer Society*: 1150.
- Lowe, D. G. (2004). "Distinctive Image Features from Scale-Invariant Keypoints." *Int. J. Comput. Vision* 60 (2): 91-110.
- Luhmann T., Robson S., Kyle S. and Hartley I. (2006). *Close Range Photogrammetry Principles, Methods and Applications*, Whittles Publishing Country, United Kingdom.
- Luigi, A. G. (2009). MyCrustOpen. File Exchange - Matlab, .
- Lysak, Y. and O. Kapshiy (2008). A dense stereo matching by iterated dynamic programming with using Gabor filters. *Modern Problems of Radio Engineering, Telecommunications and Computer Science, 2008 Proceedings of International Conference on*.
- Madsen K., H.B. Nielsen and O. Tingleff (2004) "Methods for Non-Linear Least Squares Problems."
- Manferdini, A. and F. Remondino (2010). Reality-Based 3D Modeling, Segmentation and Web-Based Visualization. Digital Heritage. M. Ioannides, D. Fellner, A. Georgopoulos and D. Hadjimitsis, Springer Berlin Heidelberg. 6436: 110-124.
- Mason, S. (1995). "Expert system- Based Design of Close-Range Photogrammetric Networks." ISPRS Journal of Photogrammetry and Remote Sensing 50(5): 13-24.
- Matlab (2010). *Optimization Toolbox User's Guide* Mathworks, Inc.
- Mattei, R. d. (2008). "The Virtual Museum of Iraq." The Italian National Research Council from <http://www.virtualmuseumiraq.cnr.it/prehome.htm>.
- McGlone, J. C., E. M. Mikhail and J. Bethel (2004). *Manual of Photogrammetry*. Bethesda, Maryland, United States of America, American Society for Photogrammetry and Remote Sensing.
- Mehra, R., P. Tripathi, A. Sheffer and N. J. Mitra (2010). "Technical Section: Visibility of noisy point cloud data." *Comput. Graph.* 34(3): 219-230.
- Mena-Chalco, J. P. (2010). Ray/box Intersection. Matlab file exchange, MathWorks Inc. Massachusetts, U.S.A.
- Meshlab (2010). *Visual Computing Lab - ISTI - CNR*, <http://meshlab.sourceforge.net/>.

- Möller, T. and B. Trumbore (1997). "Fast, Minimum Storage Ray/Triangle Intersection." Journal of Graphics, gpu & game tools 2 (1): 21-28.
- Nister, D. (2001). Automatic Dense Reconstruction from Uncalibrated Video Sequence PhD. Dissertation, Stockholm University.
- Olague, G. (2002). "Automated Photogrammetric Network Design Using Genetic Algorithms." Photogrammetric Engineering and Remote Sensing 68 (5): 423-431.
- Olague, G. and E. Dunn (2007). "Development of a Practical Photogrammetric Network Design Using Evolutionary Computing." The Photogrammetric Record 22(117): 22-38.
- Olague, G. and R. Moher (1998). Optimal Camera Placement to Obtain Accurate 3D Point Positions. International Conference on Pattern Recognition.
- Photometrix (2003). iWitness. <http://iwitnessphoto.com/index.html>
- Photoscan, A. (2011). "AgiSoft StereoScan." <http://www.agisoft.ru/>
- Photosynth, M. (2012). Photosynth. <http://photosynth.net/default.aspx>
- Pierrot-Deseilligny, M. (2012). "MicMac, software for automatic matching in the geographical context." <http://www.micmac.ign.fr/index.php?id=6>."
- Pix4D (2013). Hands free solutions for mapping and 3d modeling. <http://pix4d.com/>.
- Pollefeys, M., L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops and R. Koch (2004). "Visual Modeling with a Hand-Held Camera." Int. J. Comput. Vision 59 (3): 207-232.
- Quan, L. (2010). Image-Based Modeling, Springer Science Business Media.
- Rao, S. S. (2009). Engineering Optimization: Theory and Practice. Canada, John Wiley & Sons, Inc.
- Remondino, F. and S. El-Hakim (2006). "Image-based 3D modelling: A review." Photogrammetric Record 21(115): 269-291.
- Remondino, F., S. F. El-Hakim, A. Greun and L. Zhang (2008). "Turning Images into 3-D Models " IEEE Signal Processing Magazine 25 (4): 55-64.
- Remondino, F., F. Menna, A. Koutsoudis, C. Chamzas and S. El-Hakim (2013). Design and implement a reality-based 3D digitisation and modelling project Digital Heritage 2013 Int. Congress, IEEE Proceedings, 3D ICONS.
- Remondino, F., S. Pizzo, T. Kersten and S. Troisi (2012). Low-Cost and Open-Source Solutions for Automated Image Orientation – A Critical Overview. Progress in Cultural Heritage Preservation. M. Ioannides, D. Fritsch, J. Leissner et al, Springer Berlin Heidelberg. 7616: 40-54.
- Remondino, F. and A. Rizzi (2010). "Reality-based 3D documentation of natural and cultural heritage sites—techniques, problems, and examples." Applied Geomatics 2(3): 85-100.

- Saadatseresht, M., C. S. Fraser, F. Samadzadegan and A. Azizi (2004). "Visibility Analysis in Vision Metrology Network Design." *The Photogrammetric Record* 19 (107): 219-236.
- Saadatseresht, M., F. Samadzadegan and A. Azizi (2005). "Automatic Camera Placement in Vision Metrology Based on a Fuzzy Inference System." *Photogrammetric Engineering & Remote Sensing* 71 (12): 1375-1385.
- Saadatseresht, M. and M. Varshosaz (2007). "Visibility prediction based on artificial neural networks used in automatic network design." *Photogrammetric Record* 22(GEOBASE): 336-355.
- Shi, J. and C. Tomasi (1994). Good Features to Track. *IEEE Conference on Computer Vision and Pattern Recognition*. Seattle.
- Shin, D.-H., H. Yoo, C.-W. Tan, B.-G. Lee and J.-J. Lee (2008). "Occlusion removal technique for improved recognition of partially occluded 3D objects in computational integral imaging." *Optics Communications* 281(18): 4589-4597.
- Samantha 3DFLOW (2012). "Structure and Motion" from <http://www.3dflow.net/technology/samantha-structure-from-motion/>
- Smits, B. (1998). "Efficiency Issues for Ray Tracing." *journal of graphics tools* 3(2): 1-14.
- Snavely, N. (2008). Scene Reconstruction and Visualization from Internet Photo Collections Doctoral thesis, University of Washington.
- Snavely, N. (2010). "Bundler: Structure from Motion (SfM) for Unordered Image Collections. <http://phototour.cs.washington.edu/bundler>".
- Stanco, F., S. Battiato and G. Gallo (2011). Digital Imaging for Cultural Heritage Preservation. New York, USA.
- Tola, E., V. Lepetit and P. Fua (2008). A Fast Local Descriptor for Dense Matching proceedings of computer vision and pattern recognition (CVPR), Alaska, USA.
- Tomasi, C. and T. Kanade (1992). "Shape and Motion from Image Streams under Orthography: a Factorization Method." *International Journal of Computer Vision* 9 (2): 137-154.
- Vergauwen, M. and L. V. Gool (2006). "Web-based 3D Reconstruction Service." *Mach. Vision Appl.* 17(6): 411-426.
- Vicon. (2010). "Boujou." from <http://www.vicon.com/boujou/>.
- Virtual and Hampson. (2010). "<http://hampson.cast.uark.edu/>".
- Waldhaeusl, P. and C. Ogleby (1994). 3 by 3-Rules for Simple Photogrammetric Documentation of Architecture. *International Archives of Photogrammetry and Remote Sensing*. Melbourne, Australia 30: 426-429.
- Wang, G. and Q. M. J. Wu (2011). Guide to Three Dimensional Structure and Motion Factorization. London, Springer-Verlag Limited
- Wenzel, K., et al. (2013). "Image Acquisition and Model Selection for Multi-View stereo " *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XL-5/W1: 251-258.

- Wenzel, M. R. K. (2013). SURE - Photogrammetric Surface Reconstruction from Imagery. <http://www.ifp.uni-stuttgart.de/publications/software/sure/index.en.html>.
- Williams, A., S. Barrus, R. K. Morley and P. Shirley (2005). "An efficient and robust ray-box intersection algorithm." *Journal of Graphics, GPU, & Game Tools* 10 (1): 45-60.
- Wilson, K. and N. Snavely (2013). Network Principles for SfM: Disambiguating Repeated Structures with Local Context. *Computer Vision (ICCV), 2013 IEEE International Conference on*.
- Wolf, P. and B. DeWitt (2000). *Elements of Photogrammetry with Applications in GIS*. McGraw-Hill
- Wu, C. (2012). "VisualSFM : A Visual Structure from Motion System." from <http://homes.cs.washington.edu/~ccwu/vsfm/>.
- Yang, C. C. and F. W. Ciarallo (2001). "Optimized Sensor Placement for Active Visual Inspection." *Journal of Robotic Systems* 18 (1): 1-15.
- Yang, Y. and A. L. Yuille (1995). "Multilevel enhancement and detection of stereo disparity surfaces." *Artificial Intelligence* 78: 121-145.
- Yoeli, P. (1985). "The making of intervisibility maps with computer and plotter." *Cartographica* 22(3): 88-103.
- Zephyr 3DFLOW (2013). "Turn your photos in 3D" from <http://www.3dflow.net/3df-zephyr-pro-3d-models-from-photos/>

List of publications

- Alsadik, B., M. Gerke, and G. Vosselman (2014b). Visibility analysis of point cloud in close range photogrammetry. In: Proceedings of ISPRS Technical Commission V Symposium, 23-25 June 2014, Riva del Garda, Italy. Peer reviewed Annals, Volume II-5, 2014 / ed. by F. Remondino and F. Menna. ISPRS. ISSN: 2194-9050. pp. 9-16
- Alsadik, B., M. Gerke, G. Vosselman, A. Daham and L. Jasim (2014a). "Minimal Camera Networks for 3D Image Based Modelling of Cultural Heritage Objects." *Sensors* 14 (4): 5785-5804.
- Alsadik, B., M. Gerke and G. Vosselman (2013a). "Automated camera network design for 3D modelling of cultural heritage objects." *Journal of Cultural Heritage* 14 (6): 515-526.
- Alsadik, B., F. Remondino, F. Menna, M. Gerke and G. Vosselman (2013b). Robust extraction of image correspondences exploiting the image scene geometry and approximate camera orientation. Proceedings of 3D-ARCH 2013 : 3D Virtual Reconstruction and Visualization of Complex Architectures, Trento, Italy, ISPRS Archives.
- Alsadik, B., M. Gerke and G. Vosselman (2012). Optimal Camera Network Design for 3D Modeling of Cultural Heritage. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Melbourne, Australia, XXII ISPRS Congress. I-3: 7-12.

Summary

Image-based modelling techniques are important for producing 3D models in a practical and cost effective manner. This modelling starts with designing the camera network for the study object. This task is, however, quite crucial in practical applications because it needs a thorough planning and a certain level of experience. The optimal camera network is designed keeping the number of camera shots at a minimum.

In this thesis, we developed an automated method for designing the optimal camera network for a given cultural heritage building or statue. Starting from a rough point cloud derived from a video image stream, the initial configuration of the camera network was designed, assuming a high-resolution state-of-the-art non-metric camera. The procedure is based on reducing a dense simulated camera network. To improve the image coverage and accuracy, we used a mathematical non-linear optimization with constraints. The objective function of the optimization is based on minimizing the average error in the object space. From the experimental tests in chapter 4, we found that the target accuracy could be maintained with a minimal number of images. For the case study in chapter 7, the camera network design with a required accuracy of 10 mm is maintained from 118 images.

Furthermore, synthetic images are created to guide the camera operator to the locations where the images are to be taken. These synthetic images are created by retrieving the rough 3D model that has been produced from video imaging in the first step of design. Afterward, the real captured images are to be captured in the field in an iterative way for the whole block of images.

Then, image orientation is followed to finally generate the final 3D model. Accurate image-based models can be created as long as precise image calibration and orientation information are retrieved. The first step for orientation is to have sufficient correspondences across the captured images. Matching SIFT keypoint descriptors is a successful approach for finding these correspondences. The extraction of precise image correspondences is crucial for the subsequent image orientation and image matching steps. After the extraction of a sufficient and reliable set of image correspondences, a bundle adjustment is used to retrieve the image orientation parameters.

In this thesis, camera network design results in a combined set of short and wide baseline high resolution images covering an object of interest. Moreover, the design includes an approximation of image orientations, a rough 3D object geometry and image connectivity matrix indicating for each image its matching mates. The subsequent image matching is based on the contribution on the pre-knowledge of the image orientations and the pre-

created rough 3D model of the study object. Ultimately, the matching information retrieved during that step will be used for a bundle block adjustment.

Since the initial image orientation is defined in the design of the camera network, we can compute the image connectivity matrix beforehand. For each image involved in several pairs defined in the connectivity matrix, the keypoints are extracted. Then, its correspondences in the matching connected images are transformed by using the designed orientation and initial 3D model. Moreover, a window is defined for each keypoint and its initial correspondence in the matching images. A SIFT matching is implemented between every matching window to find the homologous points. This is followed by a bundle adjustment to orient the images automatically to finally have a sparse 3D model. The developed approach provided a sub-pixel accuracy and effective orientation when compared to the results of commercial and open source software which does not exploit the pre-knowledge about the scene.

The produced 3D model after the image orientation is to be tested whether some gaps are exists. The presented technique in chapter 6 is based on a voxelization approach. The empty voxels are tested for visibility from the captured images and an assumption is made to label them as potential gaps. A 3 by 3 neighbourhood filtering is also suggested for refinement and final gap detection. Subsequently, auxiliary images should be captured iteratively to cover the detected gap areas and a new 3D model is created as shown in chapter 7.

The developed method is aimed for cultural heritage, this will contribute to driving the development of practical, easy-to-implement digital imaging, 3D modelling and preservation solutions.

Samenvatting

Beeldgebaseerde modelleringstechnieken zijn belangrijk voor het op een handige en kosteneffectieve manier produceren van 3D modellen. Deze modellering begint met het ontwerp van een cameranetwerk rondom het object dat gemodelleerd dient te worden. Deze taak is heel belangrijk in praktische toepassingen, omdat het een grondige planning en een zekere mate van ervaring vereist. Het optimale cameranetwerk wordt ontworpen terwijl het aantal cameraposities beperkt wordt gehouden.

In dit proefschrift hebben we een geautomatiseerde methode ontwikkeld voor het ontwerpen van het optimale cameranetwerk voor het modelleren van cultureel erfgoed, in het bijzonder gebouwen en standbeelden. Uitgaande van een ruwe puntenwolk, die is afgeleid uit videobeelden, wordt een eerste cameranetwerk ontworpen voor opnamen met een hoge-resolutie state-of-the-art niet-metrische camera. De procedure is gebaseerd op het reduceren van een gesimuleerd netwerk met zeer veel camera's. Ter verbetering van de overlap tussen de beelden en de nauwkeurigheid gebruiken we een wiskundige niet-lineaire optimalisatie met voorwaarden. Doelstelling hierbij is de gemiddelde fout in de objectruimte te minimaliseren. Uit de experimenten in hoofdstuk 4 bleek dat de beoogde nauwkeurigheid met een minimaal aantal beelden kon worden behaald. Voor de casestudy in hoofdstuk 7 wordt het ontwerp van het cameranetwerk met een vereiste nauwkeurigheid van 10 mm behaald met 118 beelden.

Verder worden synthetische beelden gemaakt om de fotograaf te leiden naar de plaatsen waar de beelden moeten worden opgenomen. Deze synthetische beelden worden gegenereerd met behulp van het ruwe 3D-model dat uit de video-opnamen in de eerste fase van het ontwerp is afgeleid. Daarna worden iteratief voor het hele blok de echte opnamen in het terrein genomen.

Vervolgens worden de beelden georiënteerd om het uiteindelijke 3D-model te kunnen genereren. Nauwkeurige beeldgebaseerde modellen kunnen worden gereconstrueerd wanneer de camera's gekalibreerd en de cameraoriëntaties bepaald kunnen worden. De eerste stap voor de oriëntatie is om voldoende corresponderende punten tussen de gemaakte foto's te bepalen. Het matchen van SIFT key points is een succesvolle aanpak voor het vinden van deze punten. De bepaling van de precieze locaties van corresponderende punten is cruciaal voor de daarop volgende beeldoriëntatie en image matching. Na het inwinnen van een voldoende omvangrijke en betrouwbare set van corresponderende punten wordt een stralenbundelvereffening gebruikt om de oriënteringsparameters te bepalen.

In dit proefschrift resulteert het ontwerp van een cameranetwerk in een set van hoge-resolutie-beelden met zowel korte als lange basislijnen die het te modelleren object afdekken. Bovendien levert het ontwerp benaderde waarden van de beeldoriëntaties, een ruw 3D model en een matrix die aangeeft welke beelden met elkaar overlappen. De daarop volgende image matching maakt gebruik van de voorkennis van de beeldoriëntaties en het vooraf gemaakte ruwe 3D model van het object. Uiteindelijk worden de resultaten van de image matching gebruikt in de bundelvereffening.

Omdat de initiële beeldoriëntaties zijn gedefinieerd tijdens het ontwerp van het cameranetwerk, kunnen we de matrix met de overlapindicatie van de beelden vooraf berekenen. Voor elk beeldpaar dat volgens deze matrix een overlap vertoont worden de key points bepaald. Vervolgens worden de corresponderende punten in de overlappende beelden getransformeerd met behulp van de beeldoriëntatie uit het ontworpen netwerk en het eerste 3D-model. Bovendien wordt een venster gedefinieerd voor elk key point en de corresponderende locatie in de overlappende beelden. Een SIFT matching is geïmplementeerd tussen elk paar van corresponderende vensters om de corresponderende punten te vinden. Dit wordt gevolgd door een bundelvereffening om de beelden automatisch te oriënteren en eindelijk een 3D model te bepalen. De ontwikkelde aanpak voorziet, in vergelijking met de resultaten van commerciële en open source software die de voorkennis over de scène niet te benutten, in een sub-pixel nauwkeurigheid en een effectieve oriëntatieberekening.

Het na de beeldoriëntatie geproduceerde 3D-model moet worden getoetst op gaten. De in hoofdstuk 6 beschreven techniek is gebaseerd op een 3D rasteraanpak. Voor de lege rastercellen worden getoetst of ze zichtbaar zijn in de opgenomen beelden. Onder bepaalde aannames worden deze cellen als potentiële gaten aangemerkt. Filtering met een 3x3 venster wordt aanbevolen voor een verfijning en de uiteindelijke detectie van gaten. Vervolgens worden iteratief extra beelden opgenomen om de gevonden gaten af te dekken en een nieuw 3D model te produceren (hoofdstuk 7).

De ontwikkelde methode is bedoeld voor de documentatie van cultureel erfgoed. De methode zal dit bijdragen aan de verdere ontwikkeling van praktische en eenvoudig te implementeren oplossingen voor digitale beeldopname, 3D modellering en het behoud van cultureel erfgoed.

Curriculum Vitae

Bashar Alsadik was born in Babylon, Iraq, on 1 July 1970. From 1988 to 1992, he studied in the surveying-engineering department at Baghdad University, where he received his BSc. Degree and ranked the first.

Then, he continued with his Master of Science in the same department from 1992 to 1995 on the topic of photogrammetry. In 1999, he started lecturing in the department as an assistant lecturer. In 2005, he earned the title of lecturer and in 2008, he promoted to assistant professor. He lectured on the topics of photogrammetry and adjustment computations besides supervising M.Sc. projects.

In November 2010, he pursued a PhD research at the University of Twente, ITC faculty in the Netherlands. The research was under the theme of acquisition and quality of Geo-spatial information. His task was to automate camera network design in close range photogrammetry and 3D image based modelling of cultural heritage and the result is presented in this thesis.

ITC Dissertation List

http://www.itc.nl/research/phd/phd_graduates.aspx