3D Measurements of Buildings and Environment for Harbor Simulators
Report UMINF 09.19

Niclas Börlin∗  Christina Igasto†
Department of Computing Science
Umeå University
October 15, 2009

Abstract
Oryx Simulations develops and manufactures real-time physics simulators for training of harbor crane operator in several of the world’s major harbors. Currently, the modelling process is labor-intensive and a faster solution that can produce accurate, textured models of harbor scenes is desired. The accuracy requirements vary across the scene, and in some areas accuracy can be traded for speed. Due to the heavy equipment involved, reliable error estimates are important throughout the scene.

This report surveys the scientific literature of 3D reconstruction algorithms from aerial and terrestrial imagery and laser scanner data. Furthermore, available software solutions are evaluated.

The conclusion is that the most useful data source is terrestrial images, optionally complemented by terrestrial laser scanning. Although robust, automatic algorithms exist for several low-level subproblems, no automatic high-level 3D modelling algorithm exists that satisfy all the requirements. Instead, the most successful high-level methods are semi-automatic, and their respective success depend on how well user input is incorporated into an efficient workflow.

Furthermore, the conclusion is that existing software cannot handle the full suite of varying requirements within the harbor reconstruction problem. Instead we suggest that a 3D reconstruction toolbox is implemented in a high-level language, Matlab. The toolbox should contain state-of-the-art low-level algorithms that can be used as “building blocks” in automatic or semi-automatic higher-level algorithms. All critical algorithms must produce reliable error estimates.

The toolbox approach in Matlab will be able to simultaneously support basic research of core algorithms, evaluation of problem-specific high-level algorithms, and production of industry-grade solutions that can be ported to other programming languages and environments.

∗niclas.borlin@cs.umu.se
†Maiden name: Christina Olsén
Contents

1 Introduction ................................................. 4
  1.1 Background ............................................. 4
  1.2 Aim ..................................................... 4

2 Harbor modelling requirements .............................. 4
  2.1 Active objects ........................................... 4
  2.2 Work areas .............................................. 5
  2.3 The general area ......................................... 5
  2.4 Obstacles ............................................... 5
  2.5 Landmarks ............................................... 5
  2.6 The horizon .............................................. 5

3 Other requirements ........................................ 5

4 Literature study ............................................. 6
  4.1 Background .............................................. 6
  4.2 3D reconstruction methods — overview ..................... 6
  4.3 Sensor type and platform ................................ 7
    4.3.1 Laser scanner ......................................... 7
    4.3.2 Image-based techniques ............................... 7
  4.4 Algorithms for subproblems .............................. 8
    4.4.1 Camera calibration .................................... 8
    4.4.2 Feature point detection ............................... 10
    4.4.3 Feature point matching ............................... 11
    4.4.4 Combined feature point detection and matching ...... 11
    4.4.5 Relative orientation .................................. 12
    4.4.6 Triangulation ......................................... 13
    4.4.7 Fine-tuning (bundle adjustment) ....................... 13
    4.4.8 Densification of the point cloud ..................... 14
    4.4.9 Co-registration of point clouds ....................... 15
    4.4.10 Object extraction and model generation ............. 15
    4.4.11 Texture extraction .................................. 16
    4.4.12 Panoramic image stitching ............................ 16
  4.5 Reconstruction approaches and the type of input data .... 16
    4.5.1 Video-based reconstruction ........................... 16
    4.5.2 Reconstruction from aerial/satellite imagery ........... 16
    4.5.3 Reconstruction from laser scanner data ................ 17
    4.5.4 Image-based reconstruction ........................... 17
    4.5.5 Combination of image and laser scanner data .......... 17
  4.6 Automatic vs. semi-automatic reconstruction .............. 18
5 “Reconstruction” software
  5.1 Google Sketchup ................................. 19
  5.2 Microsoft Photosynth ............................ 19
  5.3 Photomodeler .................................. 20
  5.4 ShapeCapture/ShapeScan ........................ 20
  5.5 ImageModeler .................................. 20
  5.6 Other photogrammetric software ................. 20

6 Proof of concept .............................. 21

7 Summary and discussion ....................... 22
  7.1 Input data for harbor modelling .................. 22
  7.2 Software ..................................... 22
  7.3 Potential research areas ........................ 22
  7.4 The 3D reconstruction toolbox ................... 22

References ..................................... 24

A Sources ....................................... 33
  A.1 Journals covered ................................. 33
  A.2 Conferences .................................. 33
  A.3 Research groups ............................... 33

B Classified Reference List ...................... 34
  B.1 Feature point detection and matching ............ 34
  B.2 Camera calibration, bundle adjustment, and optimization ...... 40
  B.3 Relative and absolute orientation, 3D reconstruction, co-registration 44
  B.4 Dense stereo .................................. 48
  B.5 Interpretation, labelling, and segmentation of 3D data ........... 49
  B.6 Error analysis .................................. 51
  B.7 Applications .................................. 53

C Toolbox ...................................... 59
  C.1 Project idea ................................. 59
  C.2 Toolbox organization ........................... 59
  C.3 Toolbox themes ............................... 59
  C.4 Algorithms .................................. 60
    C.4.1 Orientation .................................. 60
    C.4.2 Triangulation ................................ 61
    C.4.3 Feature point extraction ...................... 61
    C.4.4 Least squares matching ....................... 61
    C.4.5 Algorithm validation and simulation .............. 61
  C.5 Data organization ............................ 62
  C.6 Camera models ............................... 62
  C.7 Measurement tools ............................ 62
  C.8 Visualization ............................... 63
1 Introduction

1.1 Background

Oryx simulations\(^1\) develops and manufactures real-time physics simulator for e.g. harbor environments. Among its customers are the harbors in Gothenburg, Rotterdam, Kuala Lumpur, and Shanghai. The simulators are used for education of harbor crane operators. Currently, the items within the simulator environment are hand-modelled and therefore a large amount of objects present in a harbor scene are not modelled. Furthermore, the surrounding is only introduced in a limited fashion into the simulation, resulting in a synthetic “look-and-feel”. Recently, customers have presented the desire to have more realistically looking environment simulators. This would not only be more aesthetically pleasing but only be beneficial to training and smooth the transition between the training and real-world environment.

1.2 Aim

The aim of this pilot study is twofold: 1) Survey existing algorithms and software for creating textured 3D models of objects and the surrounding environment from images and other information sources. 2) Unless a software solution is available for the harbor reconstruction problem, formulate an implementation project with the necessary capabilities. Of the general requirements we mention speed, flexibility, and error estimates: Since the crane operators are to operate real heavy equipment after training, it is of paramount importance to have reliable error estimates of the measured values that comprise their virtual training environment.

2 Harbor modelling requirements

A harbor scene has different objects with different capture requirements. Furthermore, the requirements on the captured environment differ. In this context, objects are generally considered man-made whereas the environment is not.

Objects may be classified into active objects, obstacles, and landmarks. Potential attributes to reconstruct are shape (geometry), position, and texture. The environment consists of work areas, the general area and the horizon. Attributes to reconstruct are the topography (shape and position), and texture.

2.1 Active objects

The objects with the highest requirements for geometry and texture are the active objects. Active objects are objects that can be manipulated in the simulation environment, e.g. cargo containers or pallets. However, their exact position do not need to be recovered.

\(^1\)http://www.oryx.se
2.2 Work areas

The areas with the highest requirement for topography and texture are the work areas where the active objects are to be manipulated. Examples are container storage areas or loading-unloading areas for pallets.

2.3 The general area

The general area consist of everything except the work areas. Parts of the general area may be used for transporting objects, but no manipulation of active objects generally takes place in the general area.

The exact topography of the general area do not need to be known with a high precision, and a high-quality texture is generally not need. However, in some areas, e.g. road junctions, the road markings may have to be of high quality.

Within the general area, obstacles and landmarks are placed.

2.4 Obstacles

Obstacles are objects that are not intended to be manipulated. However, they should not be bumped into during e.g. a transportation. As such, they have medium requirements on geometry and texture. Furthermore, their position should be known with medium precision. Examples of obstacles include “concrete pigs” and light towers.

2.5 Landmarks

Landmarks are buildings that an operator can use for navigation. Most buildings outside the work area are considered landmarks. The requirement for the exact position, size and texture are comparably low. However, they must still look “good enough” from the important viewpoints within the scene.

2.6 The horizon

The horizon consist of the part of the environment that is considered “far enough” away not to have to be individually modeled. However, if the real scene has an interesting horizon, e.g. a city skyline, the horizon may still be important for navigation and realism. The horizon is considered to have medium requirement for the angular position and texture.

3 Other requirements

The harbor is a busy workplace, and site access for data capture may thus be limited. Furthermore, the cost for data acquisition should not be too high. Finally, the visualization quality is especially important from select viewpoint, e.g. at the top of the work cranes and loading/unloading areas.
4 Literature study

4.1 Background

The studied literature falls mainly within the research fields of Photogrammetry, Computer Vision, and, to a lesser extent, Computer Graphics and Surveying. See Appendix A for a list of sources and Appendix B for a list of grouped references.

Photogrammetry\(^2\) has developed since the mid-1850s, originally as a technique for creating accurate topographical maps (McGlone et al. 2004, Ch. 1). Only recently, digital images have become standard input, and some 3D measurements is still performed manual on analog aerial images. Photogrammetry carries a strong statistical tradition, with error analysis and blunder detection being an integral part of most algorithms.

Surveying (or Land surveying) has historically been used longer than photogrammetry to construct maps. Surveying techniques include angle measurements between distinct points by a theodolite. Modern surveying is typically by tacheometry, where a laser theodolite can measure both angles and distances. For optimal accuracy and identification, highly reflective synthetic targets can be used. Often the theodolite is combined with a Global Positioning System (GPS) receiver for geo-referencing (Grussenmeyer et al. 2008).

Computer Vision has developed from the desire to make computers “see” (Hartley and Zisserman 2003, Foreword), i.e. to detect, measure, analyze, and understand the 3D environment. Computer Vision has a solid foundation in mathematics, especially in projective geometry and linear algebra. Many algorithms are oriented towards full automation. The interest in 3D reconstruction from the Computer Graphics area is based on the desire to capture and visualize real scenes rather than synthetic ones. The main strength of the research field lies in rendering and visualization.

4.2 3D reconstruction methods — overview

The 3D reconstruction methods presented in the literature differ in four major aspects; sensor type, sensor platform, algorithmic approach, and error treatment. The sensor type can be range-based (laser scanning, LIDAR\(^3\)) or image-based. Either acquisition mode can be terrestrial (ground-based) or aerial (airborne). The algorithmic approaches differ widely based on the input and output requirements. Finally, the methods differ in their approach to errors, from a rigorous error analysis with presented precision values in object space coordinates, e.g. m, to error analysis in image coordinates or no error analysis at all, i.e. “it looks fine”.

\(^2\)from photos—light, gramm—a—something drawn or written, and metron—to measure

\(^3\)Light Detection and Ranging, “laser radar”
4.3 Sensor type and platform

4.3.1 Laser scanner

Most laser scanners measure the time-of-flight between an emitted laser pulse and its reflection. One (“line scanners”) or two (“image scanners”) rotating mirrors enable the laser to “scan” its surrounding. In principle, the recorded time is used to calculate the coordinates of one 3D point. However, more advanced scanners exist that record multiple echos per pulse, the reflected intensity, and even color (Akca and Gruen 2007; Remondino et al. 2005; Rottensteiner et al. 2007). Laser scanners can either be terrestrial (TLS — Terrestrial Laser Scanners) or aerial (LIDAR).

The basic algorithm for 3D reconstruction with a laser scanner is (see e.g. Remondino (2006b, Ch. 1)):

1. Acquisition of a in a scanner-local coordinate system.

2. Co-registration of multiple point clouds into a common, global, coordinate system.

3. Segmentation and structuring of the point cloud, surface generation.

4. Extraction of texture data.

4.3.2 Image-based techniques

Image-based techniques are today almost entirely based on digital still and video cameras. Both types of cameras can either be single or mounted in stereo or in multi-nocular4 configurations. Airborne or spaceborne cameras are custom-built whereas many consumer digital cameras today have a high enough quality to be used for 3D measurements (Fraser and Cronk 2009). Classical aerial imagery is taken in regular patterns at high altitude (2000-5000 m) with nadir-mounted5 cameras. Some modern cameras are so called pushbroom cameras, consisting of three to four lines angled forward, nadir, and backward (McGlone et al. 2004, Ch. 8). Low-level aerial imagery can either be obtained by nadir-mounted or oblique-looking cameras mounted on an Unmanned Aerial Vehicle (UAV) or out the window of a low-flying aircraft.

In principle, all image-based techniques use the following algorithm to calculate 3D points from the input images (see e.g. Remondino (2006b, Ch. 1)):

1. Image acquisition.

2. Detection and measurement of feature points, e.g. corners, in each image.

3. Matching of feature points between images, i.e. which 2D points correspond to the same 3D point?

---

4 camera configurations with more than two cameras

5 looking straight down
4. Calculation of the relative orientation between (pairs of) images, i.e. the relative position and orientation of the camera stations at the instants when the images were taken.

5. Triangulation, i.e. calculation of object point coordinates. This will generate a “cloud” of 3D points expressed in a local coordinate system.

6. Co-registration of multiple point clouds into a common, global, coordinate system (optional).

7. Fine-tuning of calculated object points and camera coordinates (optional).

8. Point cloud densification, i.e. measurements of more points (optional).

9. Segmentation and structuring of the point cloud, surface generation.

10. Extraction of texture data.

In addition to the above steps, calibration of each camera is required to obtain high-quality results. This can be performed separately or in conjunction with the point cloud processing.

If two cameras are fixed to a stereo rig, the rig itself can be calibrated. This corresponds to determining the relative orientation between the rig-mounted cameras. If this process is performed prior to step 4 of the algorithm above, the relative orientation problem reduces to calculating the relative orientation between successive image pairs.

### 4.4 Algorithms for subproblems

#### 4.4.1 Camera calibration

The purpose of camera calibration is to calculate parameters internal to the camera. We distinguish between two different types of parameters; linear and non-linear. The most important linear parameter is the (effective) focal length, which is generally not the same value as the focal length written on the camera or stored in the image. The effect of the non-linear parameters is commonly called lens distortion, and has the effect that projections of straight lines are not straight (Figure 1). Most mathematics of photogrammetry and computer vision relies on that no lens distortion is present, or equivalently that the images or the measured coordinates are corrected for lens distortion. Such a corrected “camera” is said to be straight-line-preserving (see Figure 2). Lens distortion can only be ignored in low precision application or with cameras with very long focal lengths (>500mm).

Camera calibration is typically performed by taking multiple images of a calibration object, see Figure 3. For optimal results, camera calibration should be performed separate to the 3D reconstruction (Remondino and Fraser 2006). If that is not possible, the internal camera parameters may be estimated together with the object coordinates (“self-calibration” or “auto-calibration”) (Hartley
Figure 1: Lines straight in object space, bent by lens distortion. Left: pin-cushion distortion. Right: barrel distortion.

Figure 2: In a straight-line-preserving camera, the object point \( \mathbf{X} \), the camera center \( \mathbf{C} \), and the projected point \( \mathbf{x} \) are \textit{collinear}, i.e. on a straight line. The distance between the image plane and the camera center is known as the (effective) focal length. In this figure, the image plane is presented in front of the camera center instead.

Figure 3: Left: A image of a calibration object with artificial targets (black circles). The targets have known three-dimensional coordinates. The code rings around four of the targets are used for identification. Right: Artificial targets attached to the outside of the Destiny lab attached to the International Space Station. Image credit: NASA.
et al. 1992; Duan et al. 2008) or during the fine-tuning stage (Fraser 1997), at the cost of a reduced quality of the result.

In order to obtain useful 3D information, the camera calibration information has to be added at some stage of the reconstruction. Some algorithms only require the non-linear parameters to be known, i.e. that the cameras are straight-line-preserving (Devernay and Faugeras 2001).

4.4.2 Feature point detection

A feature point is a point or an area of an image that is likely to be found and recognized in other images. Typical feature points are corners and circular features, although many textured areas will also be good feature points. In industrial applications, artificial targets are often added to a scene. These targets provide good feature points and are sometimes coded to aid automatic identification (Fraser and Cronk 2009), see Figure 3.

Most feature point detectors are automatic — they take an image as input and generates a list of 2D coordinates where feature points have been detected. Some detectors furthermore estimate the uncertainty of each 2D coordinate, see Figure 4. In addition, each feature point may be accompanied by a descriptor that describe the surrounding of the detected point, such as the size of the feature and the dominant direction within the region containing the feature, see Figure 5. The purpose of the descriptors is to enable matching of feature points detected in different images, i.e. to enable identification of the same 3D point viewed e.g. from different distances and/or directions.

In a comparison by Remondino (2006a), the methods by Förstner and Gülch (1987) and Heitger et al. (1992) had the highest precision of the detected 2D coordinates. Other common feature point detectors include the Harris detector (Harris and Stephens 1988), SUSAN (Smith and Brady 1997), the KLT tracker (Tomasi and Kanade 1991), and SIFT (Lowe 2004). The KLT tracker is especially common in videogrammetry.

---

Figure 4: Two corners detected by the Förstner operator (Förstner and Gülch 1987) in synthetic images. The ellipses describe the uncertainty of each corner.

---

6For simplicity, this report does not distinguish between point detectors and region detectors, found in some of the literature.
4.4.3 Feature point matching

In order to extract 3D information from 2D images, a correspondence between points in different images must be established. This process is called matching. Feature points can be matched based on the image content around them or from the descriptors calculated by the feature point detector. Furthermore, if the relative orientation between two images is known, the matching can be restricted to epipolar lines (see Figure 6 (left)) rather than the whole image. Furthermore, if a third image is used, the matching ambiguities can be further reduced (Shashua 1997; Schaffalitzky and Zisserman 2002), see Figure 6 (right). Among the feature point detectors compared by Mikolajczyk and Schmid (2003), the SIFT descriptor (Lowe 2004) had the highest tolerance to changes in viewing geometry.

4.4.4 Combined feature point detection and matching

The Least Squares Template Matching (LSTM) technique performs the matching and precise location of the matches simultaneously. The basic algorithm compares patches between images while allowing a controlled geometric and radiometric deformation (Gruen 1985, 1996), see Figure 7. The LSTM algorithm is an iterative procedure that uses initial estimates of the match positions and other geometrical parameters. If the initial estimates are good and the image
Figure 6: Left: An epipolar line restricts the search for corresponding points. Potential matches outside the epipolar line do not correspond to the same 3D point. Right: Adding a third image further reduces the ambiguity. A match found in along the epipolar line of image 2 must have a corresponding match in the third image.

Figure 7: The patch (image region marked in the right image) is matched to the image to the left by Least Squares Template Matching (LSTM). Given the initial position (dashed red) in the left image, the algorithm updates the shape and position of the region (final value, solid blue). In this example, the region was allowed to deform by an affine transformation (shift, rotate, scale, and shear).

contains enough information (texture), the algorithm will converge to more precise matching coordinates, including error estimates. Otherwise, the algorithm may fail to converge, indicating a failed matching. LSTM can also use epipolar information (Baltsavias 1991), in which case the algorithm is known as Multi-Photo Constrained Template Matching. Furthermore, LSTM can be used to fine-tune the position of feature point matched by other detectors (Remondino and Ressl 2006), see Figure 8.

4.4.5 Relative orientation

The calculation of the relative orientation between different images is central to image-based techniques. A poor estimation of the relative orientation will affect the calculation of 3D object coordinates, both in terms of the object point precision and potential blunders due to incorrect matches.
Assuming that the correspondence problem has been solved, the relative orientation can be calculated from image point correspondences. The *eight-point algorithm* (Hartley 1997; Hartley and Zisserman 2003) is the most important algorithm for calculating the relative orientation for straight-line-preserving cameras. A recent important development is the *five-point algorithm* (Nistér 2004; Stewénius et al. 2006) that uses camera calibration information and is thus more stable.

If the correspondence between image points and control points (object points with known 3D coordinates) is known, the position and orientation of the image can be calculated for single images. This process is called *resection* (Grussenmeyer and Khalil 2002). Two of the most important resection algorithms are the *Direct Linear Transformation (DLT)* (Abdel-Aziz and Karara 1971; Mikhail et al. 2001, Ch. 9.3) (straight-line-preserving) and Grunert (Haralick et al. 1994; McGlone et al. 2004, Ch. 11.1.3.4) (calibrated). Recent work on calibrated resection include Schweighofer and Pinz (2006); Olsson et al. (2009).

Automatic relative orientation is still subject to active research Läbe and Förstner (2006); Remondino and Ressl (2006); Läbe et al. (2008); Frahm et al. (2009), and challenges remain for especially low-textured images or wide-baseline images with little overlap and large differences in viewpoint.

### 4.4.6 Triangulation

The calculation of 3D coordinates for an object point given corresponding points with a known relative orientation is known as *triangulation* (Hartley and Zisserman 2003, Ch. 12.2; Förstner et al. 2004, Ch. 11.2). Beside the calculated 3D coordinate, an estimate of the *error* can also be produced (Förstner et al. 2004, Ch. 11.2.7; Heuel 2004; Förstner and Wrobel 2004, Ch. 2.3.5).

### 4.4.7 Fine-tuning (bundle adjustment)

*Bundle adjustment* is the process of simultaneously estimating object points and camera position and orientations. The process has been used in photogrammetry since the late 1950:ies (Brown 1976). After a seminal paper by Triggs et al.
Figure 9: Bundle adjustment example with only two cameras, the leftmost of which is kept stationary. Left: The initial approximations of the object points (gray) and rightmost camera. Right: The final values of the object points (blue) and rightmost camera. The camera positions during the iteration process is also indicated.

(2000), it is now widely accepted within the Computer Vision community as well.

The process is iterative and require initial approximation from e.g. relative orientation and triangulation (Mikhail et al. 2001, Ch. 5.8). The initial approximation are iteratively modified until the projection of the estimated object points into the cameras at the estimated positions matches the measured image coordinates as closely as possible, see Figure 9. Besides the updated estimates of the object points and camera positions and orientations, error estimates of all estimated parameters are also produced (Förstner and Wrobel 2004, Ch. 2.2.5). Bundle adjustment is necessary in order to get high-quality results. At the same time, the method is general, and can handle any number of cameras and object points as well as other 3D geometric objects and scene constraints, e.g. that two estimated planes should be orthogonal.

Recent development in bundle adjustment include Börlin et al. (2004); Lourakis and Argyros (2005); Dickscheid et al. (2008); Lourakis and Argyros (2009).

4.4.8 Densification of the point cloud

The point cloud generated by general feature point matching is generally sparse and unevenly distributed over the images. However, once the relative orientation between two or more images is known, it is possible to use guided matching to densify the point cloud. Several “dense stereo” algorithms have been developed. The basic algorithm matches intensity variations along scan-lines of image pairs. Continuity constraints enable calculation of almost one depth value per pixel in well-textured image areas (Scharstein and Szeliski 2002; Yoon and Kweon 2008). Extension algorithms work on three or more images (Gallup et al. 2007; Seitz et al. 2006). Most algorithms are focused toward short-baseline input, i.e. situations where the camera motion between images is small, such as e.g. with video data. However, the algorithms work comparably well on wide-baseline images if the imaged regions are almost planar. Many algorithms focus on
speed and report only qualitative accuracy results. Some recent exceptions are
Seitz et al. (2006); Mordohai et al. (2007); Zhang et al. (2009).

The Least Squares Template Matching (LSTM) technique (Gruen 1985;
Baltsavias 1991) can also be used for densification. Compared to “dense stereo”
methods, LSTM generates less dense point clouds. However, an advantage is
that bad matches can be detected and excluded and precision values can be
calculated for each generated match (Grü et al. 2004; Remondino and Zhang
2006; Remondino et al. 2008).

4.4.9 Co-registration of point clouds

Image-based methods generally generate a point cloud in the same coordinate
system, since new images are added sequentially and the relative orientation
is determined from feature point matches with the existing images. However,
if two point clouds are generated without any common images, or from other
sensors such as laser scanners, the point clouds need to be co-registered, i.e.
transformed into the same coordinate system.

If three or more point correspondences between the clouds are known, the
transformation is a simple rigid-body transformation (Arun et al. 1987; Förstner
et al. 2004, Ch. 11.1.6). For optimal robustness, point correspondences between
different clouds can be determined from artificial targets in the scene (Akca
2003) or by manual assignment.

Without point correspondences, the co-registration problem is hard. Several
algorithms have been presented in the literature, either based on point-point
(or line-line) correspondences (Besl and McKay 1992; Früh and Zakhor 2004;
Barnea and Filin 2008; Boström et al. 2008; Brenner et al. 2008; Stamos et al.
2008; González-Aguilera et al. 2009) or surface matching (Gruen and Akca 2005;
Pottmann et al. 2006; Akca 2007; Bae and Lichti 2008). Other techniques are
based on global features of the point cloud (Johnson and Hebert 1999; Huber
and Hebert 2003; Bucksch and Lindenbergh 2008) or by matching with CAD
models (Rabbani and van den Heuvel 2004; Rabbani et al. 2007). Methods for
matching point clouds detected by hybrid camera-laser scanner hardware have
also been developed (Wendt 2007; Smith et al. 2008). In order to work well,
the algorithms for unorganized point clouds require good initial values (unless
known from the hardware) or require good global coverage of the scene.

4.4.10 Object extraction and model generation

If the points have been labelled in previous stages, automatically or manually,
object extraction usually consists of fitting primitives, e.g. planar surfaces or
edges, to the labelled points (Debevec et al. 1996; Gruen and Wang 1998; Koch
2005; Yang et al. 2009). Many automatic algorithms have been developed to
detect simple planar surfaces in unlabelled terrestrial and aerial laser scanner
data (Rottensteiner 2003; Yu et al. 2008; Tarsha-Kurdi et al. 2008). Other
algorithms search for known complex geometric objects, e.g. CAD models, in
point clouds generated from images or laser scanner data (Rabbani and van den
Heuvel 2004; Ferrari et al. 2006; Rabbani et al. 2007; Leibe et al. 2008). An approach that does not try to group points into geometric objects is to generate a 3-D mesh directly from the point cloud (Fruh and Zakhor 2002; Akbarzadeh et al. 2006; Gallup et al. 2007; Mordohai et al. 2007; Pollefeys et al. 2008).

Beside the actual measurements, it is also possible to add scene constraints, e.g. that two surfaces should be orthogonal, to aid the model generation (Debevec 1996; El-Hakim 2002).

4.4.11 Texture extraction
Once the camera positions and the 3D geometry of an object has been calculated, determining what part of an image that can be used as a texture map on each surface is trivial. However, in order to generate a convincingly looking textured 3D model, some practical aspects need to be considered, e.g. occlusion (Debevec et al. 1998; Ortin and Remondino 2005) and differences in lighting (Kim and Pollefeys 2008; Troccoli and Allen 2008). Furthermore, several papers have been presented that try to analyze repeated patterns in textures (Zalesny et al. 2005; Mayer and Reznik 2007; Muller et al. 2007; Wenzel and Forstner 2008). However, the problem in the general case is still unsolved.

4.4.12 Panoramic image stitching
If images are acquired by a rotating camera, no 3-D information may be inferred (Remondino and Börlin 2004). However, it is still possible to “stitch” the images together to form a panorama as described by e.g. Brown and Lowe (2007).

4.5 Reconstruction approaches and the type of input data
This section reports some of the major high-level approaches presented in the literature. The choice of approach is strongly correlated to the type of input data, and this section is thus structured by the type of input data.

4.5.1 Video-based reconstruction
The video-based approaches, exemplified by Akbarzadeh et al. (2006); Mordohai et al. (2007); Pollefeys et al. (2008); Frahm et al. (2009) are characterized by a high level of automation based on video input. The automation is successful mainly due to the small image distortion between consecutive video frames. No specific assumptions are made about the reconstructed environment and the result is usually in the form of textured 3D meshes. Several methods are able to generate automatic results in real-time, albeit without error estimates.

4.5.2 Reconstruction from aerial/satellite imagery
Methods based on aerial and/or satellite imagery are characterized by a high level of automation (Zhang and Gruen 2006). The automation is mainly achieved due to the regularity of the images capturing process. The output is usually in
the form of a 2.5D Digital Surface Map (DSM), including error estimates. A 2.5D DSM is a 2D grid of positions that describes the elevation above a vertical datum, such as mean sea level. A DSM typically cover a large area and is useful for generating topographic maps. However, due to its low resolution and that each position only has one height value associated with it, a DSM does not contain any information about e.g. building facades.

4.5.3 Reconstruction from laser scanner data

Methods based on laser scanner data, either aerial (LIDAR) or terrestrial (TLS), focus on analyzing huge point clouds (Rottensteiner 2003; Tarscha-Kurdi et al. 2007; Barnea and Filin 2008; Tarsha-Kurdi et al. 2008), mainly to detect common points for point cloud co-registration and planar surfaces for building detection. The result is usually with error estimates but without textures.

4.5.4 Image-based reconstruction

There is a multitude of papers about image-based 3D reconstruction in the literature. Some example applications include: city model generation from aerial images (Gruen and Wang 1998), modeling of Arc de Triomphe in Paris from tourist images (El-Hakim 2002), reconstruction of buildings using scene constraints (Debevec 2003), high-resolution reconstruction of cultural heritage objects (The Standing Buddhas of Bamyan) from multi-resolution images (Grü et al. 2004; Remondino and Niederoest 2004), and a fully automatic measurements for industrial applications (Fraser and Cronk 2009). As stated by Remondino and El-Hakim (2006, abstract), “... image-based modelling ... remains the most complete, economical, portable, flexible and widely used approach”. However, no unified method exist that cover all reconstruction problem. Indeed, from the same paper

...there is no single modelling technique able to satisfy all requirements of high geometric accuracy, portability, full automation, photo-realism and low cost as well as flexibility and efficiency (Remondino and El-Hakim 2006, p. 272).

4.5.5 Combination of image and laser scanner data

Methods that use a combination of images and laser scanner data are also common. Fruh and Zakhor (2003), used aerial images and laser scans of part of a city to generate a textured DSM of the ground and rooftops. A laser-scanner and camera-equipped car was used to acquire terrestrial data. The terrestrial data was used to determine the path within the DSM driven by the car and to generate texture maps of the vertical building facades.

Rabbani and van den Heuvel (2004) used laser scans to reconstruct an industrial site. Multiple scans were used to model the bulk of the site. Images were taken to complement the laser scan data in regions that were hard to reach for the laser scanner due to the crowded scene. The image data was used to aid
interpretation of the point cloud and to provide complementary measurements of edges within the scene.

In an opposite approach, Gonzo et al. (2004) used aerial images and tacheometric data to generate an overall model of an Italian castle. Detail was added to the model from terrestrial images and laser scans. The tacheometric and laser scans were considered optional in their setup, i.e. they suggest that image-only methods would be viable.

Several papers have compared tacheometry, laser scanning, and photogrammetry to reconstruct complicated buildings, e.g. castles (Remondino et al. 2005; Landes et al. 2007; Grussenmeyer et al. 2008). The consensus is that either technique has its strength and weaknesses, but that they complement each other well. However, a comparative paper by Strecha et al. (2008) challenges the consensus by posing the question on whether image-based methods can completely replace close-range laser scanning.

In a recent paper, Remondino et al. (2009) suggests the following workflow for modeling of complicated architectures:

1. Use surveying to obtain a high-accuracy reference grid.
2. Acquire low-level oblique aerial images to model the majority of the model.
3. Take terrestrial images to for complementary modelling and detailed modelling of parts of the scene.
4. Use medium-range (1–50m) laser scanner of interiors less suited for image-based modeling.

4.6 Automatic vs. semi-automatic reconstruction

A 3-D reconstruction application should ideally satisfy the following requirements (El-Hakim 2002):

1. High geometric accuracy.
2. Capturing all details.
3. Photo-realism.
4. Full automation.
5. Low cost.
6. Portability.
7. Flexibility in applications.
8. Efficiency in model size.
Thus, automation is a highly wanted requirement. However, the only available fully automated techniques either generate incomplete textured meshes without any error information (Akbarzadeh et al. 2006; Mordohai et al. 2007; Pollefeys et al. 2008; Frahm et al. 2009) or require a substantial modification of the scene (Fraser and Cronk 2009). Indeed, many authors suggest that semi-automated methods are the most efficient, and e.g. Mayer (2008, p. 217) concludes that: “Key factors determining the practical usefulness of a system are thorough testing as well as an optimized user interaction” (our emphasis).

5 “Reconstruction” software

In this section, some software related to 3-D reconstruction from images are described. Evaluated software is listed in sections 5.1–5.3.

5.1 Google Sketchup

Google Sketchup\textsuperscript{7} is a free tool for constructing 3D models that can be uploaded to the Internet and viewed by Google Earth. It can generate textured models but does not perform any measurements from images.

5.2 Microsoft Photosynth

Microsoft Photosynth\textsuperscript{8} is a free\textsuperscript{9} tool for organizing images of a scene. The application uses SIFT (Lowe 2004) features to determine the relative orientation, followed by bundle adjustment with restricted camera self-calibration to improve the estimated 3D coordinates and camera positions. The camera self-calibration is restricted to the focal length and two radial lens distortion parameters. For details, see Snavely et al. (2006, 2008).

Presently (June 2009) it is not possible to view the result on your local machine. Instead, it is necessary to upload the result to the Photosynth web site, where it is automatically made publicly available. The result is presented as a virtual 3D world, where it is possible to change the viewpoint between different calculated camera coordinates. Snavely et al. (2008, p. 191) writes

\ldots our objective is not to synthesize a photo-realistic view of the world from all viewpoints \textit{per se}, but to browse a specific collection of photographs in a 3D spatial context that gives a \textit{sense} of the geometry of the underlying scene

(their emphasis) and as of this writing it is not possible to extract 3D information from the generated result.

\textsuperscript{7}http://sketchup.google.com

\textsuperscript{8}http://photosynth.net

\textsuperscript{9}However, a Microsoft Live ID is required.
5.3 Photomodeler

Photomodeler\(^{10}\) is a photogrammetric software from EOS Systems, Inc. The basic software is based on manual measurements, although an automatic camera calibration component is included. Add-ons include automatic measurement of coded targets and dense matching. The software can export 3-D textured models as well as the raw 3-D data, including positions of object points and cameras. Furthermore, uncertainty estimates for the calculated positions are also available. The list price (June 2009) for the software is between USD 1000 (basic version) and USD 2700 (complete version).

5.4 ShapeCapture/ShapeScan

ShapeCapture and ShapeScan are two software from ShapeQuest Inc\(^ {11}\). According to the ShapeQuest homepage, ShapeCapture provides the capability for 3D Modeling from images, Camera Calibration, Accurate 3D Coordinate Measurement, Photogrammetry, Texture Mapping, Automatic Target Extraction and Stereo Matching. Furthermore, the ShapeScan software is be able to work with both images and point clouds acquired by a laser scanner. However, ShapeCapture/ShapeScan are unable to measure stored parametric models of e.g. buildings. The list prices (June 2009) for ShapeCapture and ShapeScan are USD 1600 and USD 8000, respectively.

5.5 ImageModeler

Autodesk ImageModeler is a reconstruction software sold by Autodesk Inc\(^ {12}\). According to the Autodesk homepage, ImageModeler is able to reconstruct photo-realistic objects, scenes and cities from images as well as take measurements of buildings. The software can export the results in FBX, Maya, OBJ, and DWG formats. However, the FAQ\(^ {13}\) does not mention anything about extracting the precision of the calculated values. The list price (June 2009) of the software is USD 1500.

5.6 Other photogrammetric software

Other photogrammetric software include Australis\(^ {14}\) and iWitness PRO\(^ {14}\), V-STARS\(^ {15}\), and DPA-Pro\(^ {16}\) oriented mainly at industrial applications.

---

\(^{10}\)http://www.photomodeler.com  
\(^{11}\)http://www.shapecapture.com  
\(^{12}\)http://www.autodesk.com/imagemodeler  
\(^{13}\)http://images.autodesk.com/adsk/files/imagemodeler2009_faq0.pdf  
\(^{14}\)http://www.photometrix.com.au  
\(^{15}\)http://www.geodetic.com  
\(^{16}\)http://www.aicon.de
Figure 10: Reconstruction results from Fors Nilsson and Grundberg (2009). Top left: Manual “painting” of image regions corresponding to different object surfaces. Top right: Resulting point cloud labelling. Bottom row: Two textured views of the reconstructed result.

6 Proof of concept

A proof of concept for 3D measurements of buildings from image data is presented in the work by Fors Nilsson and Grundberg (2009). The reconstruction is based on planar surfaces and uses calibrated cameras. Initially, the acquired images are ordered sequentially by the operator. Feature detection and matching is performed automatically between consecutive images and the relative orientation and triangulation is calculated using robust methods. The triangulation result is presented to the operator for quality control. Adequate results are added to the sequential registration process. Otherwise, the process is re-run with different parameters. At any stage, bundle adjustment may be run to fine-tune the reconstructed data.

Object measurements is also semi-automatic. The operator marks image parts that correspond to the same planar object surface with a “paint” tool. Labelling of the reconstructed point cloud is automatically inferred from the images, and the planes are calculated from the labelled points. The result is presented to the user that has the possibility to repeat the calculation after removing mis-labelled or wrongly calculated points. Finally, several planes are combined into a polyhedral object. See Figure 10 for an example.

In agreement with Mayer (2008), the report stresses the importance of optimizing the workflow for the user and that in some cases a too high degree of automation may be of more harm than good. As future work the authors suggest fitting points to other surface primitives than planes. Furthermore, reconstructed composite primitives could be stored in a “library” and later retrieved when a similarly composed object is to be reconstructed. Other suggestions include detail enhancement on facades with dense stereo methods and guided matching to increase the number of matched surface feature points.
7 Summary and discussion

7.1 Input data for harbor modelling

Some harbor scenes cover a substantial area, and conventional aerial methods carry the advantage of being able to cover large areas quickly. However, aerial data also carry a substantial cost. Furthermore, it is believed that on some sites, obtaining the proper permissions may be difficult or even impossible. Obtaining low-level aerial data from e.g. a UAV (unmanned aerial vehicle) could be easier and cheaper, but the maturity of the technology is unclear. However, an interesting possibility is that the harbor cranes provide elevated vantage points that may be utilized instead of low-level aerial images.

Instead, we argue that we should focus on terrestrial images and possibly laser scanner data. A possible solution is to model the general area from terrestrial image data. The work areas could either be modelled from image data or from short-range laser scanner data. An interesting prospect would be to use high-resolution images acquired by a calibrated stereo rig, something that increases the robustness of e.g. relative orientation and triangulation.

7.2 Software

There is no software that satisfy all our requirements. Of the investigated software, ShapeCapture/ShapeScan (Section 5.4) seem to be the best candidate for the harbor reconstruction problem. However, the software lacks the possibility to construct parameterized 3D models of e.g. buildings that can later be retrieved and used to speed up measurements of repeated structures. Furthermore, there is a question mark on the quality of the dense point clouds generated by ShapeScan from image data.

7.3 Potential research areas

Among the identified potential research areas within this field are

- Camera calibration, especially stereo rig calibration.
- Bundle adjustment, especially bundle adjustment with scene constraints.
- Optimization of camera networks.
- 3D reconstruction for physical simulation.

7.4 The 3D reconstruction toolbox

We suggest that a toolbox is implemented with algorithms for 3D reconstruction. The main data source should be terrestrial images and laser scan data and optionally tacheometry data. State-of-the-art automatic low-level algorithms for well-understood subproblems should be implemented “as is”. However, algorithm without error estimates should be avoided unless they can be
augmented to include error estimates. Higher-level algorithms, automatic or semi-automatic, can be combined from the low-level algorithms and evaluated to find efficient measurement workflows.

The toolbox is suggested to be implemented in a high-level language, Matlab\textsuperscript{17}. Matlab supports efficient implementation of many automatic 3D reconstruction algorithms. Furthermore, it has enough support for GUI implementations to enable the necessary efficiency evaluations. Additionally, non-GUI algorithm can be ported with relative ease to efficient implementations in low-level languages such as C/C++. More toolbox details are presented in Appendix C.

It is our belief that a 3D reconstruction toolbox in Matlab will be a flexible tool since it can simultaneously support basic research of core algorithms, evaluation of problem-specific high-level algorithms, and production of industry-grade results that can be spawned and ported to other environments. The development will be incremental, and by using the toolbox in different 3D reconstruction projects it is possible to learn from different cases.

A potential application of the toolbox on the harbor reconstruction problem would be to model the ground as a 3-D topographic mesh and reconstruct it from image data and possibly laser scanner data. Ground texture data would automatically be extracted from the images. Distant landmarks with restricted image coverage could be modelled using scene constraints or stored composite primitives as suggested in Section 6. Composite primitives could also be used for obstacle modelling and modelling of active objects. Whether the geometric quality of the reconstructed models is sufficient for modelling of active objects remains an open question and will be interesting to investigate. The horizon could finally be reconstructed from image data acquired from e.g. the harbor cranes using panoramic image stitching as described in Section 4.4.12.

\textsuperscript{17}http://www.mathworks.com
References


29


A Sources

A.1 Journals covered


A.2 Conferences

- XXI International CIPA Symposium, 01-06 October 2007, Athens, Greece.

A.3 Research groups

B Classified Reference List

B.1 Feature point detection and matching


B.2 Camera calibration, bundle adjustment, and optimization


B.3 Relative and absolute orientation, 3D reconstruction, co-registration


B.4 Dense stereo


B.5 Interpretation, labelling, and segmentation of 3D data


### B.6 Error analysis


**B.7 Applications**


C Toolbox

C.1 Project idea

• Input: A set of overlapping images of buildings and optionally 3D data from a laser scanner.

• The buildings are not assumed to be free-standing, i.e. the toolbox must be able to handle real-world restrictions such as occlusions, etc.

• The toolbox should be able to handle images taken by uncalibrated or calibrated camera, optionally on a stereo rig.

• The toolbox should support measurements of primitives such as points, lines, and planes.

• It should be possible to combine primitives into e.g. a rectangular house with a saddle-back roof.

• It should be possible to use stored combined primitives for faster measurements of recurring objects.

• The output should be a geometric model with precision estimates and texture maps for visualization.

C.2 Toolbox organization

• Key questions:
  – What algorithms should be included?
  – How should the data be organized?
  – What kind of user interface is required? (API/GUI)
  – What about data import/export features?
  – Compatibility with other software?

C.3 Toolbox themes

• The algorithms should be as automatic as possible.

• The user input should be where it is most efficient, i.e. either at the beginning or the end of a calculation pipeline.

• If a choice is necessary, the code should be transparent, i.e. easy to read, rather than optimized.

• The code should contain references to the literature.

• Estimation functions should produce error estimates in addition to the estimated values.
• For non-advanced users, the estimation functions should have traffic-light features (green=ok, red=not ok, yellow=manual attention required).

Quality measures (Fraser) 0th...3rd

C.4 Algorithms

Tentative list of core algorithms:

• Orientation
• Triangulation
• Bundle adjustment
• Camera calibration
• Feature detection
• Feature matching
• Least squares matching
• Lens distortion correction
• Co-registration
• Texture extraction

Other algorithms:

• Coded targets
• ... 

C.4.1 Orientation

• Relative orientation
  – Direct solution
    * Eight-point
    * Seven-point
    * Five-point
  – Statistically optimal solution
  – Robust solutions

• Trifocal tensor
  – Direct solution
    * Eighteen-parameters
* Eleven-parameters
  – Statistically optimal solution
  – Robust solutions

• Absolute orientation
  – Direct
  – Optimal

C.4.2 Triangulation
• Forward intersection
• Bundle adjustment
  – Datum
  – Control points
  – Self-calibration
  – Functional (hard) constraints
  – Soft constraints
  – Robustness (bad initial values)
  – Robustness (outliers)
  – Robust estimation (Huber, Cauchy, ...)

C.4.3 Feature point extraction
• Förstner/Köthe interest operator.
• Harris corner
• SIFT

C.4.4 Least squares matching
• Standard
• Multi-photo constrained

C.4.5 Algorithm validation and simulation
• Simulation
  – Camera networks
  – Image distortion
• Optimization of camera networks
• Algorithm validation
C.5 Data organization

Tentative data organization:

- Project
  - Cameras
  - Images
  - Measurements
  - Objects
  - Uncertainties
  - ...

C.6 Camera models

- Perspective (pin-hole)
  - Outer orientation
    * roll, pitch, yaw
    * azimuth, tilt, swing
    * other
  - Lens distortion models
    * Brown K1, K2, K3, P1, P2
    * Chebychev polynomials
    * Modelling of zoom lenses
    * ...

- Non-perspective cameras
  - Line-cameras
  - Fish-eye
  - ...

- Stereo rig

C.7 Measurement tools

- GUI for
  - Manual measurements
  - “Guided” measurements
  - Input of approximate values
C.8 Visualization

- Points
- Lines
- Cameras
- Residuals
- Error ellipsoids
- ...