3D Challenges and a Non-In-Depth Overview of Recent Progress

Luc Van Gool[1,2] Bastian leibe[1] Pascal Müller[1] Maarten Vergauwen[2] Thibaut Weise[1]

[1]*D-ITET / BIWI*
Swiss Federal Institute of Technology (ETH)
Sternwartstrasse 7, 8092 Zürich, Switzerland
vangool@vision.ee.ethz.ch

[2]*ESAT / PSI-VISICS*
Katholieke Universiteit Leuven
Kasteelpark Arenberg 10, 3001 Leuven, Belgium
Luc.VanGool@esat.kuleuven.be

Abstract

Although a lot of effort already went into the development of 3D acquisition technology, and existing methods come of age, several challenges remain. We try to give a – probably incomplete – overview of these. Then, some of our recent work at ETH Zürich and the University of Leuven is discussed, where we try to tackle such outstanding issues.

1. 3D scanning solved?

The production of 3D models has been a popular research topic already for a long time now, and important progress has been made since the early days. Nonetheless, the community is well-aware of the fact that still much remains to be done. In this paper we list some of these challenges (in this section), and in subsequent sections we describe recent work at ETH Zürich and K.U.Leuven which attempts to tackle them, at least in part,.

There is a wide variety of techniques for creating 3D models, but depending on the geometry and material characteristics of the object or scene, one technique may be much better suited than another. For example, untextured objects are a nightmare for traditional stereo, but too much texture may interfere with the patterns of structured-light techniques. Hence, one would seem to need a battery of systems to deal with the variability of objects to be modeled. Unfortunately, for quite a few objects, for instance in a typical museum, none of the existing techniques will work particularly well, or at least a combination thereof would be required.

As a matter of fact, having to model the entire collection of diverse museums is a useful application area to think about, as it poses many of the challenges, often several at once. In our own work, the modeling of cultural heritage has been one major driving force. Even in this demanding context, claims have been made that 3D scanning is a solved problem, but we would definitely consider such claims premature. Another area is 3D city modeling, which has quickly grown in importance over the last years. Such models can help planners, but also increasingly car drivers with navigation. In a way, it is another extreme in terms of conditions under which data have to be captured, in that streets represent an absolutely non-secluded environment, at a much bigger scale than what one would typically expect in a museum.

Here is a list of challenges that we see for such applications, which we don't claim to be exhaustive:

- Many objects have an intricate shape, the scanning of which requires great precision combined with great agility of the scanner to capture narrow cavities and protrusions, deal with self-occlusions, fine carvings, etc.
- The types of objects and materials that potentially have to be handled are very diverse, ranging from metal coins to woven textiles; stone or wooden sculptures; ceramics; gems in jewellery and glass. No single technology can deal with all these surface types and for some of these types of artifacts there are no satisfactory techniques yet developed.
- The objects to be scanned range from tiny ones like a needle to an entire landscape containing petroglyphs or cities. Ideally, one would handle this range of scales with the same techniques and similar protocols.
- For many applications, data collection may have to be undertaken on-site under potentially adverse conditions, transporting ruggedised equipment to remote sites.
- Objects are sometimes too fragile or valuable to be touched and need to be scanned 'hands-off'. The scanner needs to be moved around the object, without it being touched, using portable systems.
- Masses of data often need to be captured, like in our museum collection or city modeling examples. Efficient data capture and model building is essential if this is to be practical.

- Those undertaking the digitisation may or may not be technically trained. Not all applications are to be found in industry, and technically trained personnel may very well not be around. This raises the need for intelligent devices that ensure high quality data through (semi-)automation and strong operator guidance.
- Cultural artifacts often have huge intrinsic value. However the straight economic benefits often accrue to organisations who are not the owners or managers of the assets (e.g. in hotels, restaurants, etc. rather than at the excavations or sites discovering or safeguarding the heritage). In practice the money that can be spent is usually very limited within memory institutions and solutions for this kind of application areas therefore need to be relatively cheap. At least for a long time, a similar situation existed for 3D city models, where public authorities had high hopes for such models, but low budgets.
- Also, precision is a moving target in many applications and as higher precisions are obtained, new applications present themselves that push for even higher precision. Analysing the 3D surface of paintings to study brush strokes is a case in point.

These considerations about the particular conditions under which models may need to be produced, lead to a number of desirable, technological developments for 3D data acquisition.

Combined extraction of shape and surface reflectance. Increasingly, 3D scanning technology is aimed at also extracting high-quality surface reflectance information. Yet, there still is an appreciable way to go before high-precision geometry can be combined with detailed surface characteristics like full-fledged BRD (Bidirectional Reflectance Distribution) or BTF (Bidirectional Texture Function) information.

In-hand scanning. The first truly portable scanning systems are already around. But the choice is still restricted, especially when also surface reflectance information is required and when the method ought to work with all types of materials, incl. metals. Also, transportable here is supposed to mean more than 'can be dragged between places', i.e. rather the possibility to easily move the system around the object, optimally by hand. But there also is the interesting alternative to take the objects to be scanned in one's hands, and to manipulate them such that all parts get exposed to the fixed scanner. This is not always a desirable option (e.g. in the case of very valuable or heavy pieces), but has the definite advantages of exploiting the human agility in presenting the object and in selecting optimal, additional views.

On-line scanning. The physical action of scanning and the actual processing of the data often still are two separate steps. This may create problems in that the completeness and quality of the data can only be inspected after the scanning session is over. It may then be too late or too cumbersome to take corrective actions, like taking a few additional scans. It would be very desirable if the system would extract the 3D data on the fly, and would give immediate visual feedback. This should ideally include steps like the integration and remeshing of partial scans. This would also be a great help in planning where to take the next scan during scanning.

Opportunistic scanning. Not a single 3D acquisition technique is currently able to produce 3D models of even a large majority of exhibits in a typical museum. Yet, they often have complementary strengths and weaknesses. Untextured surfaces are a nightmare for passive techniques, but may be ideal for structured light approaches. Ideally, scanners would automatically adapt their strategy to the object at hand, based on characteristics like spectral reflectance, texture spatial frequency, surface smoothness, glossiness, etc. One strategy would be to build a single scanner that can switch strategy on-the-fly. Such a scanner may consist of multiple cameras and projection devices, and by today's technology could still be small and light-weight.

Multi-modal scanning. Scanning should not only combine geometry and visual characteristics. Additional features like non-visible wavelengths (UV,(N)IR) have to be captured, as well as haptic impressions. The latter would then also allow for a full replay to the public, where audiences can hold even the most precious objects virtually in their hands, and explore them with all their senses.

Semantic 3D. Gradually computer vision is getting at a point where scene understanding becomes feasible. Out of 2D images, objects and scene types can be recognized. This will in turn have a drastic effect on the way in which `low'-level processes can be carried out. If high-level, semantic interpretations can be fed back into 'low'-level processes like motion and depth extraction, these can benefit greatly. This strategy ties in with the opportunistic scanning idea. Recognising

what it is that is to be reconstructed in 3D (e.g. a car), can help a system to decide how best to go about, resulting in increased speed, robustness and accuracy.

Off-the-shelf components. In order to keep 3D modeling cheap, one would ideally construct the 3D reconstruction systems on the basis of off-the-shelf, consumer products. At least as much as possible. This does not only reduce the price, but also lets the systems surf on a wave of fast-evolving, mass-market products. For instance, the resolution of still, digital cameras is steadily on the rise, so a system based on such camera(s) can be upgraded to higher quality without much effort or investment. Moreover, as most users will be acquainted with such components, the learning curve to use the system is probably not as steep as with a totally novel, dedicated technology.

Obviously, once 3D data have been acquired, further processing steps are typically needed. These entail challenges of their own. Improvements in automatic remeshing and decimation are definitely still possible. Also solving large 3D puzzles automatically, preferably exploiting shape in combination with texture information, would be something in high demand from several application areas. Level-of-detail (LoD) processing is another example. All these can also be expected to greatly benefit from a semantic understanding of the data. Curvature alone is a weak indicator of the importance of a shape feature in LoD processing. Knowing one is at the edge of a salient, functionally important structure may be a much better reason to keep it in at many scales.

In the following, we present several recent approaches from our research groups at ETH Zürich and the University of Leuven, by which we try to address some of those challenges. Section 2 describes a fast acquisition system, with on-line rendering of scanned data, which we use as a basis for our planned in-hand scanning system. Section 3 discusses early implementations of the semantic 3D idea for city modeling. Finally, Section 4 describes a system requiring the absolute minimum of apparatus at the user's side: a digital still or video camera, a PC, and an Internet connection. This gives her free access to tools that turn uploaded images into 3D models.

2. Towards on-line, in-hand 3D Scanning

In this section, we describe our ongoing work to create a system that combines in-hand scanning with on-line 3D shape extraction. The system is based on the well-known phase-shift technique (see e.g. [24]). Our implementation generates 3D patches at 17 Hz. A difficulty with phase-shift is that it cannot be directly combined with in-hand scanning, as relative motions between the object and the scanner cause ripple artifacts to appear in the captured shapes.

This is a pity, as fast scanning would be especially useful with dynamic scenes. Hence, we have proposed countermeasures to strongly reduce this effect.

In order to solve the phase unwrapping problems posed by the phase-shift approach, we have combined it with stereo. It combines the accuracy of the former with discontinuity robustness of the latter. In that respect the system has a bit of the aforementioned opportunistic scanning flavour.

As is usual with phase-shift methods, surface texture can be extracted together with its shape. High-speed is achieved by implementing most of the algorithms on the GPU.

Our scanner is described in detail in [23]. Here we outline its main features. **Fig. 1** shows the components. The scanner consists of a modified DLP projector, two high-speed monochrome cameras and a color camera. By removing the color wheel of the projector, three independent monochrome images are projected by the red, green and blue color channel. The monochrome cameras are synchronized and record the three images. The color camera is synchronized as well and integrates over the three projected images. It yields the texture for the model. The setup is similar to that used by Zhang and Huang [24], but has an additional fast camera for the aforementioned stereo capacity. An example scanned with our system is shown in **fig. 2**.

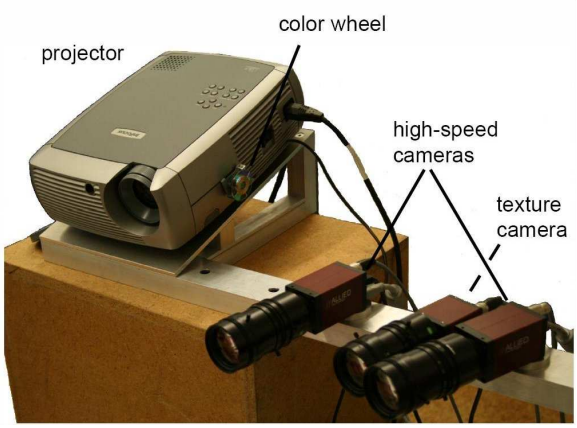


Figure 1. Hardware Setup of the on-line scanning system (more information is given in the text).



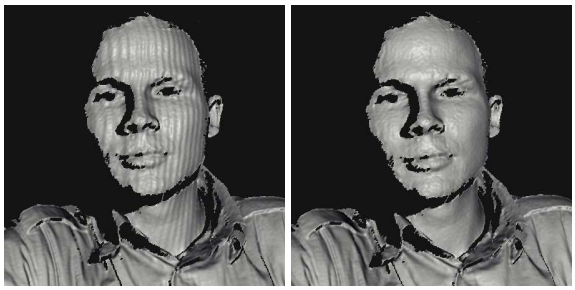
(a) (b)

Figure 2. Face reconstruction: a) geometry, b) textured geometry.

Phase-shift & Stereo. Three phase-shifted sinusoidal patterns are projected and the wrapped phase is calculated at each pixel from the intensities of the recorded images. Masking operations are used to remove all uncertain phase values due to saturation, signal strength and phase derivative variance.

The absolute phase (period + wrapped phase) is calculated using the stereo cameras. For each period hypothesis the correlation between the two monochrome cameras can be calculated using SSD. The period with highest correlation is used for the absolute phase. Period assignment errors are minimized by using Loopy Belief Propagation that optimizes smoothness of the absolute phase taking into account phase and texture borders. The absolute phase at each pixel allows to reconstruct the 3D position using point-surface triangulation between one monochrome camera and the projector.

Motion Estimation & Compensation. The phase-shift method assumes a static scene for the three recorded images. Using the DLP projector, the total acquisition time can be reduced to 14 ms. Despite this high speed, motion will still lead to artifacts. Assuming a locally planar and uniform region, however, distortion is equivalent to using a different phase-shift. This difference can be estimated locally and can thus be compensated for [23]. As a side effect, an estimate of the motion along the surface normal can be calculated at each pixel, thus providing an estimate of the 3D optical flow. Further work is necessary to see how on-line registration could benefit from this estimate (also see next point). **Fig. 3** shows a result without (left) and with (right) motion compensation activated.



(a) (b)

Figure 3. Face reconstruction a) without and b) with motion compensation. Note the ripple artifact without compensation. This was due to head motion during the scanning.

In-hand Operation. As already described in section 1, it is highly desirable to produce 3D scans swiftly, and to have an immediate impression of the completeness and quality of the model that is being produced. An interesting scenario would be to let the user manipulate the object in front of a fixed scanning setup, filter out the hands from the data, and to show a model which is gradually being completed as scanning proceeds. We are currently working on these steps. Skin color detection can leave out the hands, where after we apply crude on-line registration as described in [18,10]. This yields a model for inspection, which is then improved upon in an off-line step. For that step all scans need to be registered anew, and jointly this time [11,16], before being integrated into a complete final model [4]. Surface reflectance properties can then be recovered using the final model and the many texture images acquired during the entire scanning process.

3. Semantic 3D

Texture-mapped 3D city models are becoming increasingly important for a variety of applications. Many approaches have therefore been proposed to create such models, using 3D measurements from aerial imagery (e.g. [8]) and/or from survey vehicles equipped with laser scanners and cameras (e.g. [2,7,5,21]). Still, bottom-up reconstruction - whether from cameras or other sensors - is naturally limited in what it can infer about complex shapes like that of buildings. Clearly, 3D reconstruction can be helped by extra knowledge about the objects, *in casu* both of the architectural features of the buildings themselves, as of the objects that surround them.

3.1. Cognitive loops to the rescue

The main idea behind the system presented in the following is therefore to supply reconstruction with such semantic knowledge through the integration with and the feedback from visual object detection [3,12]. This interaction benefits from recent advances in object recognition [5,13], which have resulted in dramatic improvements in recognition performance, making such an integration finally a feasible option.

Our approach is based on the tight integration of two components. On the one hand, we employ a real-time passive-stereo based 3D City Modeling algorithm which is able to build compact 3D representations of cities using the assumption that building facades and roads can be modeled by simple ruled surfaces [2]. A typical result of this method can be seen in **fig. 4** (b). The main advantage of this algorithm is its exceptional speed. It can process the full Structure-from-Motion (SfM) and dense reconstruction pipeline at 25-29fps. Thus, the reconstructed model can directly be created online, while the survey vehicle is driving through the streets.

However, the simplifying assumptions which deliver both speed and robustness for the facade reconstructions, produce nonsensical results for other objects which defy them. As a good case in point, the algorithm is unable to model cars which are omnipresent in cities. Their textures get simply squashed onto the ruled surfaces used to represent the facades and the ground. This results in a serious degradation of the visual quality of the 3D city model (**Fig. 4** (a,b)).

We therefore combine the 3D reconstruction with the object detection algorithm from [5] in order to detect cars in the input video streams (**Fig. 4** (d)). The two components are integrated in a *cognitive feedback loop*. The 3D reconstruction modules inform object detection about the scene geometry, which greatly helps to improve detection precision. Cars are expected to be on the ground plane after all... Using the knowledge of camera parameters and scene geometry

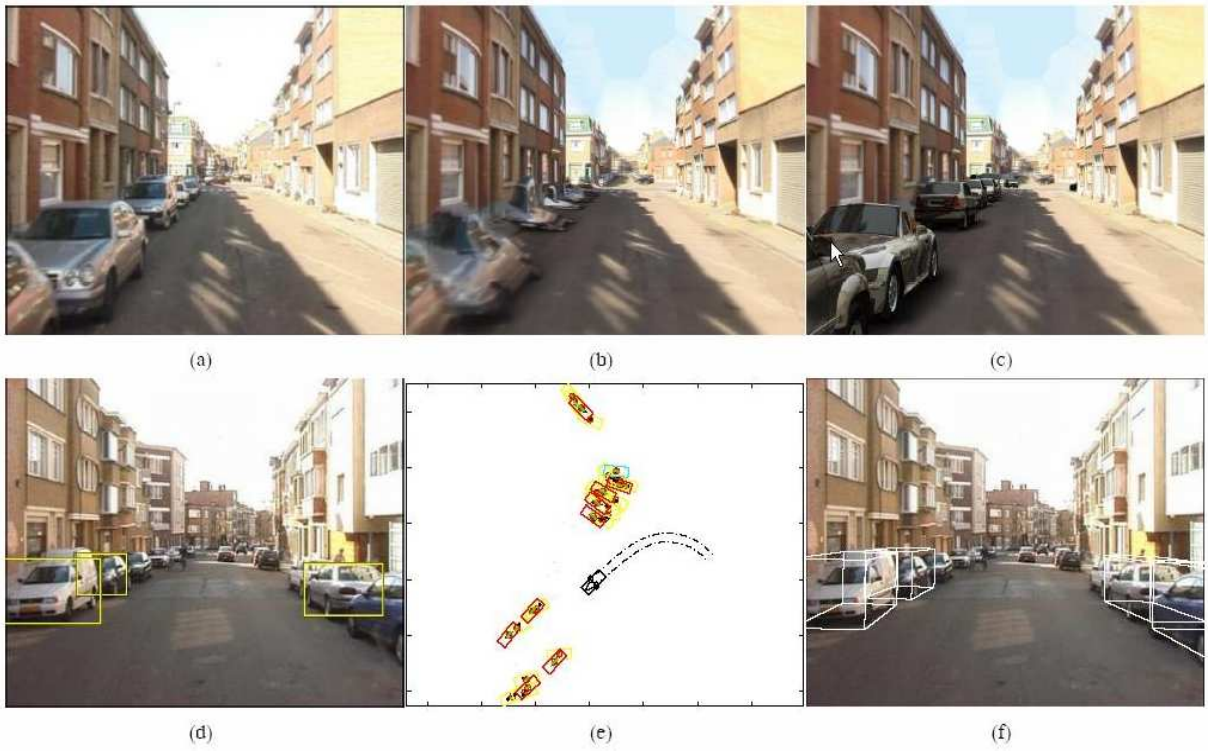


Figure 4. 3D city modeling using semantic information: (a) an image from the original survey video; (b) a rendered image from the reconstructed 3D model with the same camera position; (c) final 3D city model with virtual 3D cars whose positions have been determined by the object recognition module. (bottom) Detailed steps of the object recognition pipeline: (d) initial detections using ground plane constraints; (e) temporal integration on reconstructed map; (f) estimated 3D car locations, rendered back into the original image. They serve to instantiate the virtual cars in the final 3D city model.

from [2], the 2D car detections are temporally integrated in the world coordinate frame, leading to precise 3D location and orientation estimates (**Fig. 4**(f)). Those can then be used to build up a metric scene model (**Fig. 4** (e)) and to instantiate virtual 3D car models which improve the visual realism of our final 3D city model (**Fig. 4** (c)).

Our final system is able to create an automatic 3D city model from the input video streams of a survey vehicle, identify the locations of cars in the recorded real-world scene, and replace them by virtual 3D models in the reconstruction [3].

Besides improving the visual realism of the final 3D model by covering up reconstruction artifacts, the proposed placeholder models have several additional advantages. Since they are instantiated in the same locations as their real-world counterparts, they give a better impression of the scale of the reconstructed model and the width and passability of its streets, which is important for future, 3D car navigation systems. In addition, this solution also addresses privacy issues by removing car textures with legible license plates.

Finally, our 3D city modeling approach results in very compact models. The reconstructed city mo del for an entire test sequence (1275 stereo image pairs covering 6 streets with a total length of approximately 500m), including all facade textures, takes up only 712kB. Each placeholder car model requires an additional 300-500kB of storage, but it can be reused wherever the car is instantiated in the reconstruction.

3.2. Procedural modeling

If model size can be sacrificed for visual quality, we can also exploit similar ideas to the facades. Indeed, the characteristic features of buildings can be captured quite effectively through the use of shape grammars [14]. They do not only allow to create virtual building models faster, but also to guide the construction of realistic models of existing models from images of facades [15].

As display capabilities improve and audience expectations grow, *procedural modeling* is becoming an increasingly important supplement to traditional modeling approaches. On one side, procedural modeling allows for an efficient high-level modeling of detailed high-quality 3D content at low cost, and on the other side, the underlying mechanisms that encode the design knowledge can be exploited for image understanding in computer vision. In this section, we will first describe the specific *shape grammar* that we have constructed and employed for the procedural modeling of diverse architectural content (see **fig. 5**), and afterwards, we will present ideas and challenges of



Figure 5. This figure shows the application of *CGA shape*, a novel shape grammar for the procedural modeling of computer graphics architecture. First, the grammar generates procedural variations of the building mass model using volumetric shapes and then proceeds to create facade detail consistent with the mass model. Context sensitive rules ensure that entities like windows or doors do not intersect with other walls, that doors give out on terraces or the street level, that terraces are bounded by railings, etc.

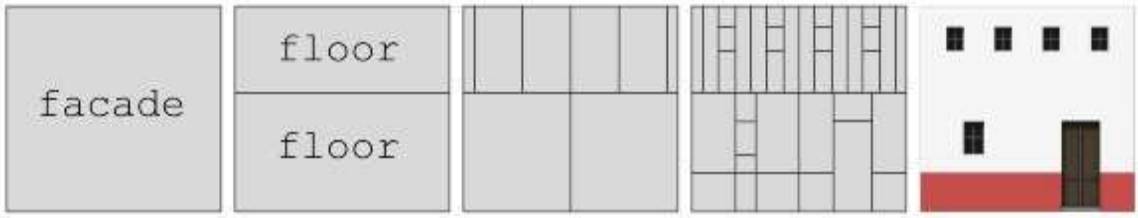


Figure 6. First 4 steps of a shape grammar derivation sequence. On the right the result of the derivation.

using such a shape grammar for the automatic extraction of semantics out of imagery, and to subsequently use it for the creation of visually appealing 3D models from monocular facade images.

3D Modeling with Shape Grammars. A landmark in the formal theory of architecture was the introduction of shape grammars by Stiny [20]. These were shown to cover a wide range of architectural styles. However, Stiny's original shape grammar was hardly amenable to computer implementation. Thus, we have proposed a more computer oriented alternative [14], *CGA Shape*, a novel attributed shape grammar. In the following, we briefly introduce the main concepts, give an example, but refer the reader to [14] for a more comprehensive description.
The idea of shape grammars is to manually define rules that iteratively evolve a design by creating more and more details. For example, the rules first create a crude volumetric model of a building, called the mass model, then continue to structure the facade and finally add details for windows, doors and ornaments (see **fig. 6**).

CGA Shape. The CGA Shape framework consists of (1) the shape definition, (2) the production process, (3) the rule notation with shape operations suited for architecture, and (4) an element repository:

Shape: A shape consists of a symbol (string), geometry (e.g. polygonal mesh), and attributes. The most important attributes are the position **P**, three orthogonal vectors **X**, **Y**, and **Z**, describing a local coordinate system, and a size vector **S**. These attributes define an oriented bounding box in space called *scope*.
See **fig. 7**.

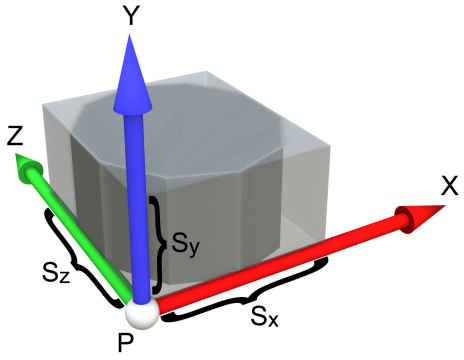


Figure 7. Shape with scope.

Production process: The production process can start with an arbitrary configuration of shapes, called the initial shapes, and proceeds as follows: (1) Select an active shape with symbol in the set (2) choose a production rule which acts on the active shape and replaces it by a set of successor shapes, (3) mark the initial shape as inactive and add the successor shapes to the configuration and continue with step (1). The resulting data set is called *shape tree*.

Rules: The CGA Shape production rules are defined in the following form:

predecessor : *condition* → *successor* : *prob*

predecessor is a symbol identifying a shape that is to be replaced with *successor*, and *condition* is a guard (logical expression) that has to evaluate to true in order for the rule to be applied. The rule is selected with probability *prob*. Several different types of shape operations can be applied to specify the successor shape, e.g. transformations to modify the scope, to split faces, to repeat structures, etc. Dimensions can be specified in absolute or relative terms.

Element repository: The library of 3D models consists mainly of basic primitives and elementary architectural objects (e.g. ionic capitals) created with traditional modeling tools like Autodesk's Maya. They are hierarchically organized in categories and types and each element has a unique identifier, shader attributes and optional metadata.

Virtual reconstruction example. As an example, we describe how the ruined site of Pompeii can be reconstructed with CGA Shape. To create such 3D model, we have to rely mainly on the footprints and the available architectural knowledge about the building designs of that epoch.
Hence, the first task was to study extensively the domestic architecture of Pompeii (e.g. [6,17,22]). This resulted in specifications like: one or two-storey buildings with a height between 5m and 9m existed, the lower parts of facades are often painted in a redish color, remarkably large doors (4 meter) are prevalent, the windows were small and barred, etc.
The next task was to classify the different building appearances. We ended up having three types of building designs: (1) shops with large openings, (2) two-storeyed hotels with a more elaborate door decoration etc, and (3) simple, one-storey domestic houses. Based on such archaeological knowledge, real building footprints (=initial shapes) and GIS data which drives the rule selection (building type), ancient Pompeii was reconstructed with 190 manually written CGA shape rules. The whole model is a rule-based

composition of just 36 element objects. A rendering of the reconstruction is depicted in **fig. 8**.

Including probabilities in the rules is interesting when one wants to create models from little information as in the Pompeii case. If an archaeological excavation has brought to light little more than the footprints of a building, and otherwise one only has information about the style in which the building had been erected, then one can quickly generate multiple reconstructions, all equally consistent with the foundations and the style. Each reconstruction applies the same rule set, but rules are applied with certain probabilities. So, rather than building a model with explicit indications of uncertainty, one can draw multiple samples (i.e. reconstructed models) from the family of possibilities, show these, and in that way convey an impression of what is (close to) certain and what is rather speculative.



Figure 8. Virtual reconstruction of Pompeii.

Modeling facades. In the context of city modeling, an extra, important step has to be taken. Starting from images, and given a shape grammar for the buildings, select a set of rules (and the corresponding parameters) and leaf nodes (basic templates, e.g. for window types), so that the facade texture (or shape in case a 3D reconstruction is available) can be re-created or 'explained' with these rules.

In order to create such models, a kind of reverse engineering process is needed. This is a challenging task, and results so far have been piecemeal. In [1], the authors demonstrated, in a simplified scenario, that it is possible to apply a Bayesian approach to automatically determine the control parameters of a grammar.

We have recently proposed a system that yields such grammatical models from single images of facades [15]. This is the kind of information that we also obtain from the fast city modeling approach of the previous section. Given a single rectified image of a building facade as input (rectification is possible from vanishing points), we address the problem of automatically computing a 3D geometric model that (1) looks like a plausible interpretation of the input image, (2) has much higher resolution and visual quality than the input image, and (3) includes a semantic interpretation (with known windows, doors, storeys, etc.).

The proposed approach works as follows. First, mutual information is used to extract high-level facade structure by detecting repetitions. Afterwards, a subdivision scheme determines semantic top-down hierarchies via synchronized edge detection. And finally, the resulting shape tree can be used to infer a shape grammar rule set. **Fig. 9** shows an example. On the left a low-quality input image is shown, on the right the enhanced version (incl. higher resolution, windows lying deeper, etc.). This transition was achieved fully automatically.



Figure 9. Left: input image of low quality. Right: enhanced results exploiting semantic understanding of the façade structure.

By figuring out the *meaning* of the parts, their relative position with respect to the main plane of the facade can be inferred. The relative depth of the windows, for instance, will not be perfect in this way, but the rendering of deeper windows with added specular reflection by the glass, markedly increases the realism.

Current image-based modeling methods for architecture are either inaccurate or need a lot of manual input. We believe that the addition of semantic information to the building modeling process will become standard procedure.

4. 3D WebService

One of the issues mentioned in section 1 was cost. It was suggested that the use of consumer hardware can help in keeping it at bay. Over the last couple of years we have been working on a 3D WebService. It allows users to produce 3D models at virtually no cost, once they have available some standard consumer products, which many people have these days.

4.1. Overview

Users of the 3D WebService only need a digital photo-camera, a PC, and an Internet connection. The user takes images of the object or scene to be reconstructed in 3D, uploads the images to a server, and gets notified by e-mail when the 3D results are ready for download.

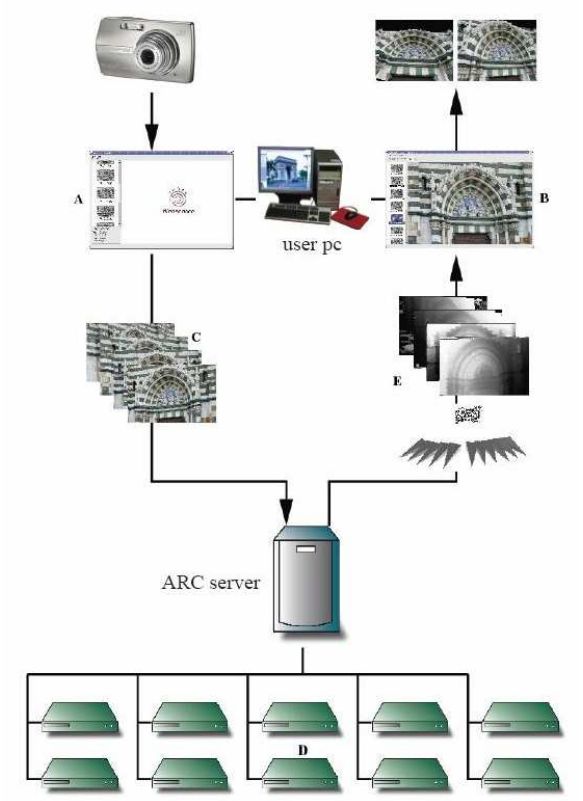


Figure 10. Schematic overview of the client-server setup. Images (C) are uploaded from the upload tool (A) on the user side to the server. There they are processed on a PC cluster (D). The results (E) can be downloaded via ftp and visualized on the user PC with the modelviewer tool (B).

Fig. 10 shows a schematic overview of the client-server setup of the 3D webservice. The client- (or user-) part is located at the top. The server side is at the bottom. On his PC, the user can run two programs, the *upload tool* and the *modelviewer tool*, indicated with **A** and **B**. In the upload tool, images can be imported that were taken with a digital camera. Once authenticated, the user can transfer these images (**C**) over the Internet to the server at ESAT (EE dept. at K.U.Leuven). There a fully automatic parallel process is launched which computes dense 3D information from the uploaded images. The parallel processes are run on two clusters of Linux computers (**D**), the first of which is a small local cluster of PC's while the second is the large workstation cluster of K.U.Leuven (800+ processors). When the server has finished processing, the user is notified by email and the results can be downloaded from the ARCserver by FTP (The name ARC stands for Automatic Reconstruction Conduit). These results consist of dense depth maps for every image and the corresponding camera parameters (**E**).

The modelviewer tool allows the user to inspect the results. Every reconstructed depth map in the image set can be shown in 3D, unwanted areas can be masked out and the meshes can be saved in a variety of formats. This modelviewer tool only provides a limited functionality. Alternatively, the *MeshLab* software of the Visual Computing Lab at CNR-ISTI (Pisa, Italy) can be used for the visualization. This freely available tool was recently extended with a plug-in that reads the results of the 3D webservice. It allows to import the results into a 3D viewer, clean them up, perform some filtering operations and merge the different depth maps into one model.

Automatic Reconstruction Pipeline. The 3D webservice is meant to create 3D reconstructions from a wide variety of images. Because no user interaction is possible once the images have been uploaded, an important prerequisite is the need for robustness and autonomy on the server part. A more detailed description of the processing pipeline is given in [25]. Next, we give a short summary.

Pipeline Overview. The processing pipeline consists of roughly four steps:

1. A step that computes a set of image pairs that can be used for matching, including Subsampling and Global Image Comparison modules. In this step, the images are first sub sampled (hence the hierarchical nature of the pipeline). Since images can be uploaded

in non-sequential order, we have to figure out which images can be matched. This is the task of the Global Image Comparison algorithm which yields a set of image pairs that are candidates for pair wise matching.

2. A step that performs the Pair wise and Projective Triplet Matching and the Self Calibration. In this step, feature points are extracted in the sub sampled images. All possible matching candidates of step 1 are now tried. Based on the resulting pair wise matches, all image triplets are selected that stand a chance for projective triplet reconstruction. This process is performed and the results are fed to the self-calibration routine which finds the intrinsic parameters of the camera.

3. A step that computes the Euclidean reconstruction and upscales the result to full resolution. In this step all image triplets and matches are combined into one 3D Euclidean reconstruction.

4. A step that is responsible for the dense matching, yielding dense, i.e. pixel-wise, depth maps for every image.

Opportunistic Pipeline. Classic uncalibrated Structure from Motion (a.k.a. Structure And Motion) pipelines make use of the fact that the set of input images is taken in a sequential manner. This helps the reconstruction process tremendously because only consecutive pairs of images must be matched for 3D reconstruction. Moreover, when the input consists of video, subsequent frames are very similar and matching is therefore relatively easy. Unfortunately, the 3D Webservice described in this paper can not rely on this assumption. Users can upload images in non-sequential order or even use images that were taken in a random fashion. The system has to actively look for opportunities to turn 2D data into 3D models. This has an impact on the matching step, the reconstruction step and the dense matching step.

Another frequently encountered problem is that of recorded scenes with parts that are dominantly planar. Traditional SaM systems run into trouble here because planar scenes give rise to ambiguities in the projective reconstruction step. In our pipeline we detect the image triplets in which only a planar part of the scene is visible. These triplets are discarded for the self calibration. Once the internal parameters are computed, the discarded triplets are picked up again because now all computations can be done in metric space and planar scenes no longer pose problems.

Hierarchical Pipeline. In general the quality and accuracy of the resulting depth maps is proportional to the size of the input images. However, computing feature points and matches on large-scale images is very time consuming and not so stable a process. That is why all incoming images are first sub sampled a number of times until they reach a typical size in the order of 1000x1000. Most of the Structure And Motion processing is performed on these sub sampled images. It is only in the final upscaling step that the result is upgraded from the low resolution to the high input resolution.

Parallel Pipeline. Several operations in the reconstruction pipeline have to be performed many times and independently of each other. Image comparison, feature extraction, pair wise or triplet matching, dense matching, etc are all examples of such operations. The pipeline is implemented as a python script which is automatically triggered by the SQL database when a new job arrives. The script has to go through several steps and every step can only be started when the previous one has finished. Inside one step, however, the processing is parallelized. This makes the processing suited for execution on the KULeuven cluster of workstations. In the current implementation, every job that arrives on the ESAT server is sent to the KUL cluster first. If no resources are available there, the job automatically returns to ESAT where it is run on our local machines.

4.2. Example results

The 3D web-based reconstruction service has been running for more than a year now. Several image sets have been uploaded to the service by various users in the cultural heritage field.

Fig. 11 shows input images of the Arc de Triomphe in Paris. For this showcase a relatively high number of images were taken, also simplified by the fact that one can walk around it. An overview of the different cameras positions is given in **fig. 12**. As can be seen, more than 100 images were combined in this case. It has to be emphasized that the service can yield good results with far fewer images though. The resulting model is shown in **fig. 13**. In order to illustrate the level of detail obtained, **fig. 14** shows a part of the reconstruction.

The WebService is accessible at http://www.arc3d.be



Fig. 11 A subset of the input images for modeling the Arc de Triomphe

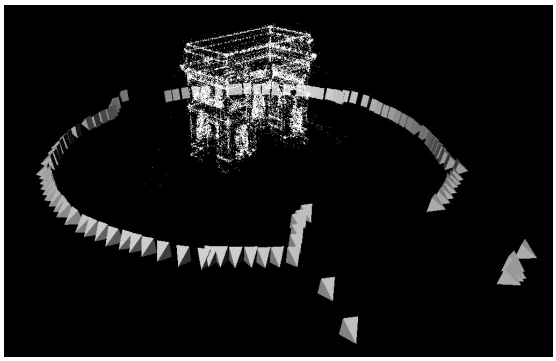


Figure 12. Overview of camera positions from where input images were taken.

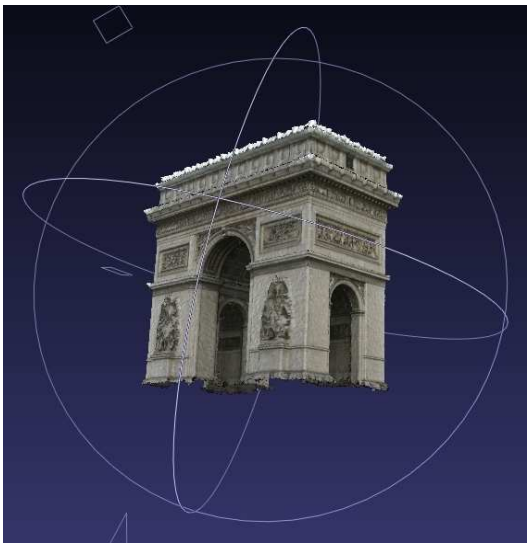


Fig. 13: resulting model of the Arc.



Figure 14. oblique view showing the level of 3D detail in one of the sculptured ornamentations of the Arc.

4.2. Conclusions and future work

There are still many problems to be solved in the area of 3D data acquisition. We have highlighted some, but there will undoubtedly be several more. This said, the community is making steady progress, and in the paper we described some of our own contributions. Due to space limitations, we had to make a selection. In [10] we have shown how structured light patterns can be

adapted on the fly to the objects to be captured (the opportunistic scanning theme). In [26] we proposed a scheme to capture detailed BTF information for rough surfaces (joint capture of 3D shapes and surface characteristics). Etc.

Acknowledgement

The authors gratefully acknowledge support from Toyota Motor Europe NV, as well as from the EC Network of Excellence EPOCH (Excellence in the Processing of Open Cultural Heritage), the EC Integrated Project DIRAC (Detection and Identification of Rare Audio-visual Cues), and the EC Training Network CHIRON (Cultural Heritage Informatics Research Oriented Network).

References

[1] F. Alegre and F. Dellaert, "A probabilistic approach to the semantic interpretation of building facades". In Int. Workshop on Vision Techniques Applied to the Rehabilitation of City Centres. 2004.

[2] N. Cornelis, K. Cornelis, and L. V. Gool. "Fast compact city modeling for navigation pre-visualization". In *CVPR'06*, 2006.

[3] N. Cornelis, B. Leibe, K. Cornelis, and L. V. Gool. "3d city modeling using cognitive loops". In *3DPVT'06*, 2006.

[4] B. Curless and M. Levoy, "A volumetric method for building complex models from range images", *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303-312, New York, NY, USA, 1996. ACM Press.

[5] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In *CVPR'05*, 2005.

[6] H. Eschebach, "Zur Entwicklung des pompeianischen Hauses," in *Wohnungsbau in Altertum* (Diskussion zur archäologischen Bauforschung 3, Berlin 1979) 152-61.

[7] C. Frueh, S. Jain, and A. Zakhor. "Data processing algorithms for generating textured 3D building façade meshes from laser scans and camera images". *IJCV*, 61:159–184, 2005.

[8] A. Gruen. "Automation in building reconstruction". In Fritsch and Hobbie, editors, *Photogrammetric Week'97*, Stuttgart, 1997.

[9] J. Hu, S. You, and U. Neumann. "Approaches to largescale urban modeling". *Computer Graphics & Applications*, 23(6):62–69, 2003.

[10] T. Koninckx, T. Jaeggli, and L. van Gool. "Adaptive scanning for online 3d model acquisition". In Workshop *on Real-Time 3D Sensors and Their Use*, Washington DC, USA, June/July 2004. IEEE.

[11] S. Krishnan, P.Y. Lee, J.B. Moore, and S. Venkatasubramanian. "Global registration of multiple 3D point sets via optimization-on-a-manifold". *Symposium on Geometry Processing*, pages 187-196, 2005.

[12] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. "Dynamic 3d scene analysis from a moving vehicle". In *CVPR'07*, 2007.

[13] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR'05*, 2005.

[14] P. Müller, P. Wonka, S. Haegler, A. Ulmer and L. Van Gool. 2006. "Procedural Modeling of Buildings". In Proceedings of ACM SIGGRAPH 2006 / ACM Transactions on Graphics (TOG), ACM Press, Vol. 25, No. 3, pages 614-623.

[15] P. Müller, G. Zeng, P. Wonka, and L. Van Gool. 2007. "Image-based Procedural Modeling of Facades". To appear in Proceedings of ACM SIGGRAPH 2007 / ACM Transactions on Graphics (TOG), ACM Press, 9 pages.

[16] K. Pulli, "Multiview registration for large data sets". In *3DIM*, pages 160-168, 1999.

[17] L. Richardson, *Pompeii. An Architectural History*. Baltimore and London 1988.

[18] S. Rusinkiewicz, O.A. Hall-Holt, and M. Levoy. "Real-time 3D model acquisition". *ACM Trans. Graph*, 21(3):438-446, 2002.

[19] I. Stamos and P. K. Allen. "3D model construction using range and image data". In *CVPR'00*, 2000.

[20] G. Stiny, G. 1975. Pictorial and Formal Aspects of Shape and Shape Grammars. Birkhauser Verlag, Basel.

[21] Y. Sun, J. K. Paik, A. Koschan, and M. A. Abidi. "3D reconstruction of indoor and outdoor scenes using a mobile range scanner". In *ICPR'02*, 2002.

[22] Andrew Wallace-Hadrill. *Houses and Society in Pompeii and Herculaneum.* Princeton University Press, 1994

[23] T. Weise, B. Leibe, and L. V. Gool. "Fast 3d scanning with automatic motion compensation". In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07),* June 2007. in press.

[24] S. Zhang and P. Huang."High-resolution, real-time 3d shape acquisition". In *CVPR Workshop*, pages 28–28, 2004.

[25] M. Vergauwen, L. Van Gool, "Web-based 3D reconstruction service", *Machine vision and applications*, vol. 17, no. 6, p. 411-426, Dec 2006.

[26] A. Neubeck, A. Zalesny and L. Van Gool, "Light Source Calibration for IBR and BTF Acquisition Setups", *Proc. 3D Data Processing, Visualisation and Transmission (3DPVT),* June 2006