A SYNTHETIC 3D SCENE FOR THE VALIDATION OF PHOTOGRAMMETRIC ALGORITHMS

Dirk Frommholz

DLR Institute of Optical Sensor Systems, Berlin, Germany - dirk.frommholz@dlr.de

KEY WORDS: Synthetic Scene, Validation, Stereo Matching, Texture Mapping, Orientation, Reconstruction

ABSTRACT:

This paper describes the construction and composition of a synthetic test world for the validation of photogrammetric algorithms. Since its 3D objects are entirely generated by software, the geometric accuracy of the scene does not suffer from measurement errors which existing real-world ground truth is inherently afflicted with. The resulting data set covers an area of 13188 by 6144 length units and exposes positional residuals as small as the machine epsilon of the double-precision floating point numbers used exclusively for the coordinates. It is colored with high-resolution textures to accommodate the simulation of virtual flight campaigns with large optical sensors and laser scanners in both aerial and close-range scenarios. To specifically support the derivation of image samples and point clouds, the synthetic scene gets stored in the human-readable Alias/Wavefront OBJ and POV-Ray data formats. While conventional rasterization remains possible, using the open-source ray tracer as a render tool facilitates the creation of ideal pinhole bitmaps, consistent digital surface models (DSMs), true ortho-mosaics (TOMs) and orientation metadata without programming knowledge. To demonstrate the application of the constructed 3D scene, example validation recipes are discussed in detail for a state-of-the-art implementation of semi-global matching and a perspective-correct multi-source texture mapper. For the latter, beyond the visual assessment, a statistical evaluation of the achieved texture quality is given.

1. INTRODUCTION

With the advent of digital processes in photogrammetry, the validation of the underlying algorithms which materialize as software packages becomes an important aspect to be systematically addressed. Any guarantees particularly on the geometric accuracy of data products derived through modern remote sensing can only be made because it is assumed that the implementations that generate these outputs and their precursors work as specified. Since formal verification is usually impractical even for small pieces of the source code of a program, rigorous tests and evaluation procedures currently remain the only option to confirm functional correctness and performance. To support the validation phase during software development, a set of reference data or ground truth is necessary which is assumed to be correct by definition. This specifically applies to photogrammetric algorithms that are used to refine image orientations and generate digital surface models (DSMs), digital terrain models (DTMs), true-ortho mosaics (TOMs), point clouds, 3D meshes and texture maps which are vital for downstream data classification and interpretation.

A few reference data sets have been published so far. The prominent and regularly refreshed Middlebury Stereo images have been used for years as a test suite and benchmark for dense stereo matching (Scharstein et al., 2014). More comprehensively, the ISPRS/EuroSDR data set of two urban scenes in Dortmund/Germany is not only being utilized as ground truth for image matching, but also for the validation of bundle adjustment algorithms (Nex et al., 2015). It includes aerial, terrestrial and close-range bitmaps as well as ground control points (GCPs) and 3D point clouds obtained from laser scanners. On a higher level, the data sets of Vaihingen/Germany, Toronto/Canada and Potsdam/Germany target classification, 3D reconstruction and 2D/3D semantic labeling (Rottensteiner et al., 2012) (Gerke et al., 2014).

Sophisticated techniques and evaluation procedures have been described and applied to these reference data sets to ensure their consistency and accuracy. This includes secondary measurements involving structured light and high-resolution aerial laser scans or mathematical approaches like plane reconstruction to filter invalid tie points. Nevertheless, the ground truth suffers from the inherent problem that real-world measurements unavoidably will be afflicted with errors from the sensing equipment due to physical limitations. Moreover, the sensor configuration that is used to capture natural scenes is fixed at the time of acquisition. As a consequence, while those data sets perfectly reflect the conditions in realistic environments and definitely are useful for comparisons on derived high-level outputs, they are less suitable for the in-depth validation and test of algorithms in early stages of the software development process. Here a controlled scene with geometric deviations in the magnitude of the respective machine precision is required to identify imperceptible functional flaws. Also, some flexibility with regard to the acquisition of the raw input data is advantageous.

To particularly address the validation of photogrammetric algorithms from the software development perspective, this paper describes the design and application of a synthetic 3D scene. The scene is constructed from differently shaped geometric objects at known positions with an accuracy in the range of the machine epsilon. Most objects are arranged inside a tiled central zone for specific tests which is surrounded by an extensive landscape. The dimensions of both areas are large enough to accommodate a variety of virtual imaging sensors and laser scanners. Although some parts may also fit close-range setups, the 3D scene predominantly focuses on aerial validation scenarios for fundamental photogrammetric algorithms. This includes bundle adjustment, dense stereo matching, the generation of digital surface models (DSMs), digital terrain models (DTMs) and true-ortho mosaics (TOMs), 3D reconstruction, automatic texture mapping (i.e. coloring polygons by assigning them parts of images), either stand-alone or as a part of

a combined workflow. Due to the exclusive use of open file formats for both the geometry and its decorations, accurate reference data like images, height maps and point clouds can be obtained from the synthetic 3D scene using freely available and commercial third-party tools. As an example, the open-source ray tracing software POV-Ray will be discussed (Persistence of Vision Pty. Ltd., 2018). To demonstrate the application of the constructed 3D scene, recipes are provided for selected validation tasks. Performance gets evaluated for a custom implementation of dense stereo matching with disparity gating and a perspective-correct texture mapping algorithm which works on unrectified input images with minimal resampling.

2. SCENE OBJECT CONSTRUCTION

At its lowest level, the synthetic 3D scene comprises a set of basic geometric objects which are grouped into larger entities. The objects are stored in boundary representation which guarantees the compatibility with current graphics hardware and application programming interfaces like OpenGL or Vulkan (Segal , Atteley, 2017)(The Khronos Vulkan Working Group, 2019). Surface-based geometry is also supported by most 3D modeling software which could have been also used to construct the entire synthetic 3D scene by hand. However, since existing interactive design tools have been found to suffer from significant limitations regarding positional accuracy, availability of sophisticated shapes and output data formats, the basic building blocks are generated by a dedicated computer program written in C++. This provides full control over the data types of the vertices, vertex references, normal vectors and texture coordinates of the object meshes. Coordinates and indices get stored as double-precision floating point numbers or long integers respectively to accommodate huge data sets and large geographic coordinates with minimal round-off errors. Moreover, the source code of the command-line oriented tool is platform-independent and aims at identical outputs on different computer systems given the same input data and settings on invocation.

2.1 Basic 2D primitives

The set of basic objects the synthetic scene is constructed from includes 2D primitives in 3-space, three-dimensional objects assembled from the 2D primitives and "organic" shapes derived from volumetric data (see figure 1). The 2D primitives comprise triangles, potentially curved quadrilaterals and sine waves. They have a surface area, but do not enclose a volume which in specific setups will affect DSM-driven algorithms. While the triangle geometry is directly generated as the connection of the three corner points, the quadrilaterals get created by bilinear interpolation of the coordinates of the four corner vertices at regularly spaced positions. The number of grid subdivisions is user-defined allowing arbitrarily smooth surfaces at the cost of a high polygon count. Vertex normals are computed from the cross-product of a linearly independent subset of three vertices and define the front and back side of the shapes. The texture coordinates of the planar triangles are derived by first aligning one edge to the x-axis of the two-dimensional texture space. The remaining vertex then gets ortho-projected onto the x- and y axes of the texture coordinate system, and the resulting coordinates subsequently are being normalized to the [0,1] range. This procedure preserves the aspect ratio, however, it cannot be applied to the curved quadrilaterals leaving some texture distortion depending on the twist.

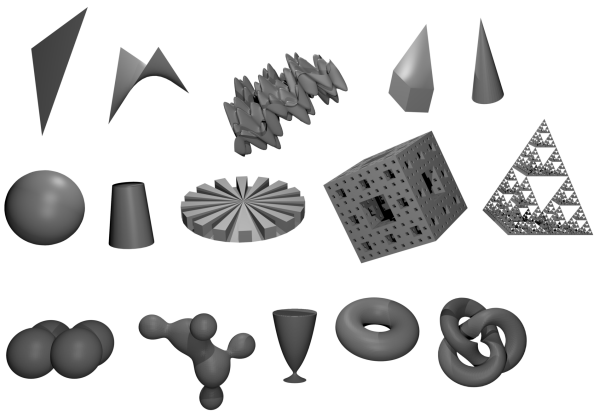

Figure 1. Basic scene objects from top left to bottom right: triangle, quadrilateral, sine wave, curved box, cone, sphere, truncated cone, Boehler star, Menger sponge, Sierpinski tetrahedron, 4- and 6-metaballs, wine glass, torus, trefoil knot.

For naturally-looking objects like mountainous landscapes or sea, additively modulated sine waves are derived from a quadrilateral grid. The coordinates of the subdivision points of its inner surface are altered according to equation 1.

$$\begin{aligned} A(x,y) =& A_0 sin(2\pi f_h x) \cdot sin(2\pi f_v y) + \\ & A_h sin(2\pi g_h x) + A_v sin(2\pi g_v y) \end{aligned} \tag{1}$$

Here A is the total amplitude by which the vertex at the integer subdivision indices x, y is moved along its normal vector. The A_0 and A_h, A_v denote the amplitudes of the base sine and the horizontal and vertical modulation functions with the respective frequencies f_h, f_v and g_h, g_v. In case negative lobes are not desired, the computed amplitudes A are replaced by their absolute value, and to seamlessly fit surrounding meshes, the border amplitude can be forced to a specific value as well. Normals of the modulated sine will get recomputed for each vertex as the cross product of the connection between the two to four neighboring vertices because they might have changed compared to the original quadrilateral.

2.2 Composite 3D objects

From the 2D primitives, genuine 3D objects like curved boxes, cones and truncated cones (with the cylinder as a special case) are assembled. Also, the Menger sponge and Sierpinski tetrahedron fractals are generated recursively from the basic shapes. These meshes have a complex but regular structure that emulates modern architecture and vegetation with challenging visibility conditions, and due to their self-similarity they qualify for resolution analysis. However, depending on the maximum recursion level, the fractals may expose a high polygon count. For the Menger sponge, in a postprocessing step, the number of faces is reduced by suppressing identical quadrilaterals during construction which will only affect the invisible inner polygons.

For a more accurate determination of the effective resolution in both the rendered images and 3D reconstructions of the synthetic test scene, the 2D primitives are further used to build Boehler stars (Huxhagen et al., 2009). The three-dimensional equivalent of the well-known Siemens star consists of elevated circular segments on a background cone where the segment width at a particular radius equals the depth to the cone coat. Thus, having a Boehler star with n elevated segments (or $\tilde{n} = 2n$ segments in total if the gaps where the background cone

shines through are included), the resolution l in its 3D reconstruction (by matching virtual stereo images or simulating a laser scanner) is

$$l = \frac{d\pi}{2n} = d\frac{\pi}{\tilde{n}} \quad . \tag{2}$$

In this equation d denotes the diameter of the circle where the regained vertices of the elevated star segments can be clearly distinguished from the background cone. This value must be obtained manually.

2.3 Complex monolithic 3D objects

Besides the objects assembled from the 2D primitives, monolithic sphere meshes are constructed by the scene generator using a modified UV approach to immediately define the north pole position without a downstream rotation. The vertices of the sphere coat are obtained by scaling the radius vector $\tilde{r} = r \cdot s$ pointing to the north by $s = cos(u\pi/(n_{stack} - 1))$ where $u \in \{0, ..., n_{stack} - 1\}$ is the current stack number, or latitude. Subsequently, at the resulting position, another vector orthogonal to the radius is constructed and scaled by $\sqrt{1 - s^2}$ to end on the spherical surface. Rotating this vector by angles of $\phi = 2v\pi/n_{slice}$ with $v \in \{0, ..., n_{slice}\}$ being the current slice number, or longitude, and adding it to the sphere center yields the final vertex coordinates. Sphere normals are constructed from neighboring vertices except for the north and south pole where they are derived from the normalized radius vector. Texture coordinates are equidistantly "unrolled" horizontally and vertically to the $[0, 1]$ range, i.e. the texture bitmap is expected to be an equidistant cylindrical projection of the sphere surface.

In addition to the rather regular objects, a few irregularly looking "organic" shapes complete the set of basic building blocks used to compose the synthetic 3D scene. This includes two sorts of metaballs (Blinn, 1982), a torus, a wine glass and a trefoil knot (Klaus, 2010) which have been chosen to particularly target tessellation algorithms. The irregular shapes are modeled as implicit surfaces as shown in general by equation 3, i.e. as a three-dimensional scalar field. For the example of the torus with the major radius R and the tube radius r, the field density is described by the quartic equation 4.

$$F(x, y, z) = 0 \tag{3}$$

$$F(x, y, z) = (x^2+y^2+z^2+R^2-r^2)^2-4R^2(x^2+y^2) = 0 \tag{4}$$

For each implicit surface, the triangular boundary representation gets approximated as the isosurface of level zero using the Marching Cubes algorithm (Lorensen , Cline, 1987). Instead of accurately unwrapping the meshes, regularly spaced texture coordinates are assigned to the resulting faces in acceptance of distortions during visualization.

2.4 Material properties

Independently of the presence of texture coordinates, each generated object is assigned at least one material. The material properties are based on the Alias/Wavefront OBJ material definitions and include ambient, diffuse and specular colors as well as a specular exponent for the shininess of the surface, a transparency value and a refraction index (Ramey et al., 1995). Also, a reference to a diffuse texture bitmap is kept which will be projected onto the surface to be colored using its normalized texture coordinates. Since the interpretation of the material properties is not standardized, render results may vary among different visualization toolsets depending on the implemented illumination models and reflectance distribution functions. Consequently, only views of the decorated objects that were produced by the same software package will be directly comparable.

For the assembled objects, each subsurface can be assigned a unique material to avoid strictly periodic patterns. Since such patterns rarely exist in a realistic environment, they would cause systematic errors with many photogrammetric algorithms operating on the synthetic 3D test scene. As an alternative to keeping multiple decorating images in memory, the subsurfaces of composite meshes can be assigned a rotation and directional scale factors to be applied to the texture coordinates when they are calculated by the generator software. By gracefully modulating these values, a sufficiently diverse appearance with just one single texture bitmap for the entire object will be ensured without consuming additional resources.

3. SCENE COMPOSITION

Using a fixed layout that is hard-coded in software, the synthetic 3D scene is automatically assembled from scaled and translated instances of the generated object meshes. It natively covers a total area of 13188 by 6144 length units and extends vertically over 4096 length units. The ground truth is subdivided into a detailed core area of 900 by 900 length units surrounded by four landscape tiles. The mesh of the entire scene comprises roughly 750k vertices and 400k polygons. Its losslessly compressed texture bitmaps occupy around 4.4 GiB of mass storage. The decoration originates from open-source real-world imagery (Kelsey, 2015) (OpenFootage, 2018) and custom-made images based on geometric patterns or coherent noise (Perlin, 1985). To lower the overall requirements on computer hardware, individual parts of the 3D model can be selectively excluded from being generated. Figure 2 shows the center of the scene for which there is also a detailed top-view map describing the exact object placement and dimensions.

Figure 2. Synthetic 3D test scene with its core area and parts of the surrounding landscape

3.1 Core area

The core area of the synthetic scene consists of eight tiles. Each of the tiles aims at the validation of algorithms for specific photogrammetric problems.

3.1.1 Central Plaza Placed in the heart of the core area, the Central Plaza is particularly designed to test dense image stereo

matchers. For this purpose, the tile of 300 by 300 length units contains differently shaped boxes, spheres, truncated cones and mutually occluded polygons positioned over a horizontal ground plane at $z = 0$. The objects are decorated with homogeneous, repeating and uniquely structured textures. To determine the spatial and vertical resolution during point reconstruction, Siemens and Boehler stars with segment counts that are fractions of $\tilde{n} = 100 \cdot \pi$ are present that render mental calculation of the achieved sample distances possible. Matching performance can further be evaluated on a three-level Menger sponge representing a visibility challenge and on Sierpinski tetrahedra with four, five and six iterations forming fields of particles of decreasing size.

3.1.2 Sine Hills To the north of the Central Plaza, the Sine Hills test area of the same size is situated. It comprises sine wave objects organized in a checkerboard pattern. The horizontal and vertical frequencies of the wave objects increase gradually along the x and y coordinate axes. Sine Hills is designed to evaluate matching penalty functions. The area also serves as a test ground for DTM extraction algorithms that are supposed to remove man-made objects and vegetation from the terrain. Here slope is a critical factor for the decision on what is to be kept.

3.1.3 Special Effects Grounds South of the Central Plaza is the Special Effects (SFX) Grounds. It comprises a planar surface on which a grid of 5 by 3 boxes is located. The boxes share the same homogeneous white material, however, in each row the values for its transparency, specularity and refractivity increase. SFX grounds is primarily designed to simulate distortions like CCD blooming in sensor simulators and to evaluate their influence on surface reconstruction. The concrete interpretation of the material properties depends on the actual image render software and its illumination model.

3.1.4 Cone Forest In its northwest corner, the Central Plaza touches the Cone Forest. This tile of the core area contains non-overlapping regular cones at random positions and with varying dimensions. Each shape is colored from a randomly assigned leaf texture bitmap loosely resembling deciduous trees and conifers. Cone Forest has been designed to validate algorithms that determine the visibility of objects and detect occlusions, either actively like in the ray casting component of texture mapping tools, or indirectly as in dense stereo matchers. To ensure that the cone configuration remains reproducible among consecutive execution runs of the 3D scene composition tool, the software cannot rely on the pseudo random number generators (PRNGs) of the C++ standard library which may differ internally among computer platforms and compilers. Therefore, a deterministic PRNG based on the linear congruential method has been implemented separately (Knuth, 1997). This PRNG is fast but not cryptographically secure, and it outputs integers only which restricts the cone coordinates and their size to non-fractional values. This limitation is due to the lack of reproducibility of floating-point numbers on different processors with varying internal arithmetics and register widths.

3.1.5 Pattern Valley Located south of the Cone Forest and west of Central Plaza, the Pattern Valley tile is dedicated to both the visual and statistical evaluation of texture mapping algorithms. It comprises a rectangular trenched open box surrounded by walls to block the view to any disturbing objects of neighboring tiles. The main area inside the box contains a set of partially levitated planar and non-planar test objects of varying surface detail. Its shapes are colored from high-contrast dichromatic texture bitmaps built from the RGB primary colors. Most bitmaps expose coarse checkerboard patterns to be accurately rasterized even when anti-aliasing is disabled in the 3D model viewer. This design, which gave the tile its name, also supports the quick visual assessment of the texture mapping outcome since any misalignment errors, color glitches and interpolated spots will catch the observer's eye immediately. Despite their coarse appearance, the textures used to dye Pattern Valley are high-resolution to suppress interpolated pixels when getting close to the objects. Moreover, to make the tile "matchable", their initially piecewise homogeneous textures have been added noise. The noise does not exceed the range of ± 10 intensity levels. By generously dimensioning the minimum and maximum values per color component for the Pattern Valley textures to avoid systematic truncation, this leaves an unused color band between 21 ... 229 intensity levels at natively 24 bits per pixel. The uniquely colored surfaces permit performance tests of a combined photogrammetric workflow, i.e surface reconstruction followed by texture mapping. Here the noise can be easily removed for the second step by applying a simple threshold filter.

Disregarding reconstruction, the outlined setup imposes a series of challenges to texture mapping tools to obtain an appealing visualization of the initially naked geometry which is to be colored. Depending on the configuration of the virtual cameras rendering samples from inside the Pattern Valley box, the software under test must properly choose and combine those candidate bitmaps which shall be utilized as texture sources, for instance by resolution and coverage. Because of the levitated objects of the tile, visibility may have to be detected in 3-space. Also, since some test bodies come with a high number of small and badly proportioned polygons, potential color inconsistencies near the face boundaries are likely to occur and need to be resolved. Moreover, the face count may increase the size of the commonly generated texture atlases which aggregate the color information for multiple scene objects in a single image. This problem is further compounded by any perspective distortions relative to the position of the virtual camera.

In contrast to its main area, the northern part of Pattern Valley has been designed to visually assess the performance of texture mapping on non-opaque surfaces. It contains two groups of objects with increasingly transparent and refractive materials. The special test room lies behind a divider that separates it from the rest of the Pattern Valley tile. The divider comprising vertically aligned polygons has been decorated with Siemens stars on both sides. This is to provide test patterns for texture resolution measurements.

3.1.6 Box Cities Northeast and east of the Central Plaza, the Upper Box City and Lower Box City tiles consist of regularly positioned axis-aligned boxes on a sine wave and quadrilateral ground surface respectively. Each box is colored with unique urban façade textures, and some of them have been added roof-like structures. Emulating single-family homes and highrise buildings, the Box Cities are supposed to serve as a simple testbed for the evaluation of 3D building reconstruction algorithms. Further, the tall and tightly packed structures of the business district of Lower Box City induce plenty of occlusions. This is specifically relevant for the simulation of virtual oblique aerial cameras and 2D urban mapping scenarios.

3.1.7 Organic City South-east of Central Plaza, the core area of the synthetic 3D scene contains a collection of organi-

cally-looking complex 3D geometries with unique monochromatic textures. This tile called Organic City comprises one instance of each object from section 2.2. It has been designed to test 3D tessellation algorithms facing irregular shapes with smoothly and sharply changing surfaces. Both remote sensing and close-range approaches for the triangulation can be evaluated from outside and inside the meshes since the monolithic objects of Organic City are hollow.

3.2 Outer rim landscapes

The core area of the synthetic scene is surrounded by four landscape tiles. They consists of sine wave objects with different modulation frequencies and amplitudes. The meshes are textured from bitmaps of 8192 x 8192 or 8192 x 4096 pixels resembling sand dunes, grassland and rocky mountains. The landscape tiles primarily serve as padding for virtual aerial cameras when the core area of the 3D scene is approached during simulated flight campaigns. Therefore, the texture resolution on the ground of about 0.75 m is adequate, and smoothing during the process of rasterization will generally be acceptable.

To also be able to validate bundle adjustment algorithms, a set of up to 5000 true 3D ground control points (GCPs) can be optionally placed inside the landscape by the 3D scene creation software. The GCPs are auto-generated and randomly but reproducibly positioned at integer coordinates using the same PRNG as for the Cone Forest from section 3.1.4. They consist of horizontally aligned quadrilateral meshes of a user-defined extent and get dyed from a texture depicting the GCP number and a Secchi-style midpoint marker (figure 3). For reference, the set of GCPs is output as a text file describing their ID and exact location within the landscape tiles.

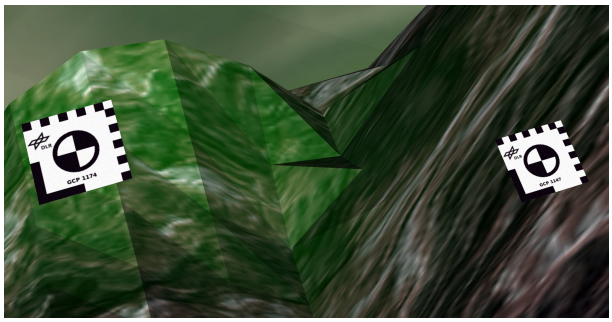

Figure 3. Randomly placed GCP markers

4. GENERATION OF DERIVED DATA PRODUCTS

The boundary representation of the synthetic scene and its material definitions are stored in the open Alias/Wavefront OBJ data format. The OBJ output of the polygonal model is human-readable, directly supported by many 3D visualization and editing tools for postprocessing and can be straightforwardly parsed by custom software. Following triangulation, the scene generator also exports the data set as a description for the POV-Ray ray tracer.

Both the OBJ and POV-Ray output allow deriving image samples and further data products from the scene meshes. The ray tracer is particularly suitable for this purpose since it internally works with double-precision numbers. Thus, it remains accurate when a validation scenario requires the scene to be translated by large offsets. Such offsets occur for instance with UTM coordinates that typically are in the $10^5 ... 10^6$ range. With single-precision numbers, in the worst case, only one bit will be left for the fractional part (IEEE Computer Society, 2008).

POV-Ray implements various lighting options and camera models, and its scene description language (SDL) is Turing complete. With a set of include files to be plugged together as needed, the tool allows the definition of virtual cameras to render pinhole and orthographic images of the synthetic 3D world with nearly no programming effort. By invoking the trace macro from inside the SDL code, it is also possible to obtain congruent perspective depth maps, DSMs and 3D point clouds (figure 4). Further, source code has been written to produce orientation files for the original semi-global matching (SGM) tool chain (Hirschmüller, 2008) and the SURE software (nFrames GmbH, 2019), which involves coordinate system transformations. Like for the 3D model, for fault injection tests or to simulate realistic conditions, the rendered outputs can be intentionally distorted in a postprocessing step, i.e. by introducing systematic errors or adding noise.

```
// loop over all image pixels (px;py)
#for (py, 0.5, image_height)

  // vertical position of pixel in world units
  // pixels grow downwards, pixvector in world grows upwards
  #local vPixWld=(0.5-(py/image_height))*UP_VECTOR;

  #for (px, 0.5, image_width)

    // compute horiz. position of pixel in world units
    #local hPixWld=(px/image_width-0.5)*RIGHT_VECTOR;

    // build, rotate, translate pixel vector
    #local pixWorld=<hPixWld, vPixWld, DIRECTION_VECTOR>;
    #local pixWldRot=vrotate(pixWorld, <
        degrees(CAMERA_ROT_ANGLES.x),
        degrees(CAMERA_ROT_ANGLES.y),
        degrees(CAMERA_ROT_ANGLES.z)
    >);

    // ray trace!
    #local intersctn=trace(model, camPos, pixWldRot, normal);

    // write depth value if intersection has been found
    #if (vlength(normal)!=0)
      #write(FILE, str(vlength(intersctn-camPos), 10, 16), " ")
    #else
      #write(FILE, DEPTH_NAN, " ")
    #end
  #end // for px

  #write(FILE, "\n")
#end // for py
```

Figure 4. POV-Ray code snippet to create ASCII depth images

5. VALIDATION RECIPES

To demonstrate the application of the synthetic 3D scene, sample workflows for the validation of common photogrammetric algorithms will be described next. Actual results will be given for a custom state-of-the-art implementation of semi-global matching with disparity gating similar to (Rothermel, 2016) and the texture mapping tool l3tex+ from (Frommholz et al., 2017).

5.1 Dense stereo matching

To assess the performance of dense stereo matching using the synthetic 3D scene, a set of two oriented RGB images and the corresponding perspective depth maps must be initially obtained, preferably from a parallel rig of virtual cameras. When using POV-Ray as a render tool and running the trace macro for each pixel, the bitmaps can be obtained simultaneously in

one render pass per camera. To suppress resampling artifacts from high-frequency image content, anti-aliasing should be enabled for rendering since it won't affect the explicitly written depth calculation loop. If a non-parallel camera setup is chosen, prior rectification is required for the RGB images for most matchers which will warp the bitmaps to have parallel epipolar lines as with the parallel camera rig. In this scenario, the depth maps have to be generated in a separate render pass using the adjusted orientation data. Any no-data areas inside the RGB outputs must be propagated respectively.

Having the inputs, the RGB data can be passed to the stereo matcher. The matcher will output a pair of disparity maps encoding the scene depth as the parallax between corresponding pixels. For a direct comparison to the ground truth, to consider occlusions correctly, the depth maps produced by POV-Ray and other render tools must be converted to disparities first since they natively store raw distances relative to the camera position. Obtaining the pixel shift and labeling mutually invisible parts of the scene involves 3D point reconstruction, reprojection and consistency checks for each depth map pixel. These calculations must be implemented separately and have been realized as a software prototype written in C++ for the test below. When both the actual and ground truth pairs of disparity maps are available, the evaluation of matching performance and correctness consists in a pixel-wise comparison of the parallax values and the calculation of error statistics.

The outlined validation recipe has been applied to RGB stereo images of 6000 x 6000 pixels of the Central Plaza tile using a custom dense stereo matcher. The software is based on the SGM algorithm with disparity gating. Census cost has been aggregated along 17 equiangular paths with fixed penalties for disparity changes, and no subsequent interpolation has been conducted. After comparing the ground truth disparities to the matching output, an average deviation of 12.25 has been found for the left image taking only the non-occluded pixels into account (32.486.270 of 36 million pixels, or 90.24%). The respective median deviation has been calculated as 0.145, and about 5.9% of the pixels showing mutually visible objects have a disparity error greater or equal than 1.0. The vertical resolution at the coarse Boehler star in the middle of the left disparity image is about 1.1 length units at a distance to the camera of 272.21 length units. Figure 5 depicts the ground truth and matching results. Because small distance values translate to a large parallax, brightness levels are inverse for the perspective depth images and disparity maps.

5.2 Texture mapping quality assessment

The Pattern Valley tile of the synthetic 3D scene has been particularly designed for the statistical evaluation of the performance of texture mapping algorithms beyond a purely visual and hence biased assessment. For this purpose, in order to re-texture the flawlessly colored original geometry, the raw OBJ mesh will be used which initially must be stripped off its textures. This can be done, for instance, by removing its material file. Also, using either the OBJ or the congruent POV-Ray description of Pattern Valley, a set of oriented RGB images from randomly parametrized virtual cameras needs to be obtained from the original textured scene. Both the stripped 3D model and the image samples are subsequently fed into the texture mapping tool under test. Its output, the re-colored 3D mesh, is now rendered once more using identical settings for the virtual cameras. After noise removal (see section 3.1.5), error statistics can be

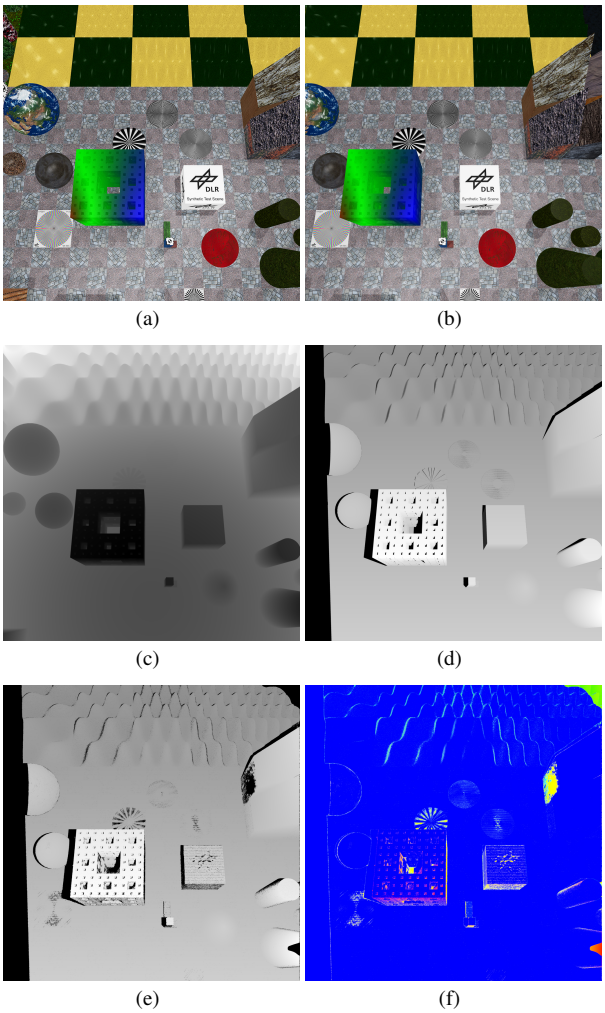

Figure 5. Stereo matching results for the synthetic 3D scene (a) Left image (b) Right image (c) Left perspective distance map from POV-Ray (d) Left reference disparity map (e) Left disparity map after matching (f) Deviation heat map

derived by comparing the resulting virtual photographs to the corresponding original oriented bitmaps pixel by pixel.

There is some important aspects to be considered when the described validation workflow is followed in the Pattern Valley area. To not make noise removal by simple thresholding impossible, the render software shall be restricted to full-intensity ambient lighting. For the same reason, if a different type of light sources is desired, the intensity at every spot of the scene tile must lie within its unused color band. Also, special care must be taken of aliasing issues which would poison the measured deviations. Because the scene will get imaged twice during the described process, any anti-aliasing or texture interpolation must be disabled within the utilized render software. However, disabling the anti-aliasing feature will produce artifacts when high-frequency textures of the 3D scene get sampled under violation of the Nyquist criterion. Therefore, as a workaround to minimize the distortion, deactivating anti-aliasing and taking close-range or high-resolution images from the synthetic 3D model is recommended for both render passes.

As a proof-of-concept for the recipe, texture mapping quality is statistically assessed for the l3tex+ software which has been previously used to texture 3D building models from oblique aerial imagery. Since l3tex+ does not ortho-rectify the polygo-

nal texture patches collected in its atlas bitmaps, neither POV-Ray nor any other freely available render tool can be used to produce the images of the synthetic 3D scene as mentioned above. Instead, a custom OpenGL-based program performs the visualization of the Pattern Valley tile. This small tool evaluates the third texture coordinate calculated by the texture mapper for the required twofold perspective correction. The correction simultaneously considers both the randomly placed cameras whose images effectively color a particular polygon and the cameras that picture the scene for the comparison process, which may be different.

For the actual test, a set of 750 randomly placed and rotated cameras with identical inner parameters has been defined. Cameras inside the 3D scene objects were discarded. Also, a DSM for the visibility test of l3tex+ was generated using POV-Ray. The resulting 24 bit RGB images of 3840 x 2160 pixels and the DSM were subsequently used to texture the polygonal mesh of Pattern Valley. Re-rendering the scene and comparing the results against the ground truth imagery revealed an average deviation of 19.14 and the expected median deviation of zero. Here both values were computed from the pixel-wise Euclidean distance within the RGB cube over the entire set of bitmaps. A more detailed analysis on the error distribution for each image pair showed that only a small fraction of the residuals originates from inaccurately aligned texture patches. For the most part, the positive average deviation results from the blind spots under the levitated scene objects which are invisible to l3tex+ due to its occlusion test based on the orthogonally-projected 2.5D surface model. Figure 6 depicts the results for a single camera position.

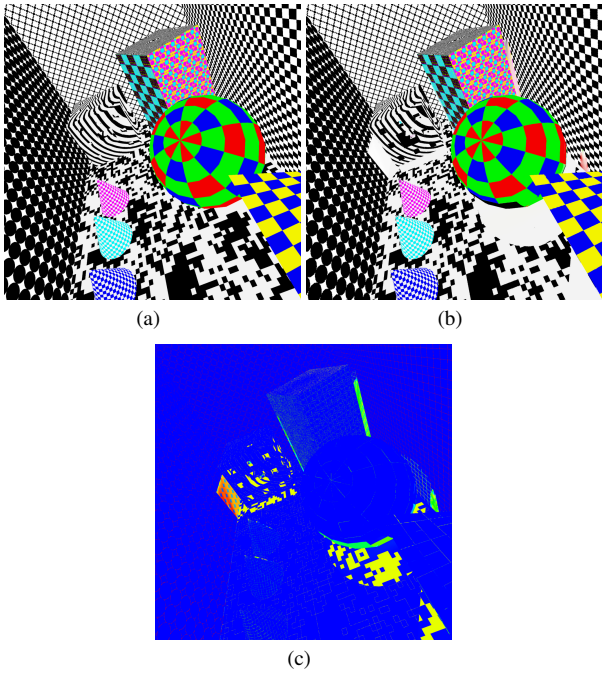

Figure 6. Texture mapping evaluation for Pattern Valley (a) Original RGB image sample (b) Image after texture mapping (c) Deviation heat map

Besides the basic validation scheme outlined above, the synthetic 3D scene can also be used to evaluate the texture mapping performance in combined photogrammetric processes, i.e. following 3D surface reconstruction. To obtain a quality measure in this case, the reference OBJ model of the synthetic scene used in the recipe initially must be replaced by the mesh restored externally. However, following noise removal in the comparison stage, oriented images of the ground truth model and its aligned and textured reconstruction must be opposed.

5.3 DSM and TOM generation

For the validation of algorithms that generate DSM bitmaps from point clouds, reference input data is obtained by ray casting the OBJ model of the synthetic 3D scene in one or more passes. Depending on the scheme, the emulation of laser scanners, linear push broom sensors or matrix cameras may be realized. Similarly, ground truth DSMs are obtained by orthogonally ray casting the scene and mapping the distance values to the nearest intersection to intensities. Both data products can be generated directly by POV-Ray using its orthographic camera and trace macro. However, simulating linear sensors this way will suffer from the disproportional I/O overhead per sample. Subsequently, the 3D point clouds are passed to the DSM software under test, and its result is compared pixel by pixel to the reference DSM to obtain the error statistics.

In order to validate TOM algorithms, oriented image data and a DSM have to be obtained from the synthetic 3D scene as a reference. Also, the TOM acting as ground truth must be generated using an orthographic projection with the designated spatial resolution. Both the oriented bitmaps and the DSM are then passed to the software under test to render its version of the true-ortho mosaic. To retrieve the respective error statistics, both TOMs are eventually aligned and compared pixel by pixel .

5.4 Automatic bundle adjustment algorithms

For the generation of test data for automatic bundle adjustment algorithms which determine the image orientations, the landscape of the synthetic 3D scene must be rendered with GCPs enabled. This will yield accurately oriented sample bitmaps and a list of coordinates of 3D reference points. GCPs of the core area can be retrieved from its commented top-view 2D map.

Subsequently, the bitmaps without their orientation and optionally a subset of the GCPs are passed to the automatic bundle adjustment algorithm. When the image orientations have been calculated, they are compared against the reference values to assess the deviation. Reprojection errors also can be calculated in particular for the GCPs. However, their pixel coordinates must be picked manually since they depend on the camera parameters used for rendering.

5.5 3D object reconstruction

The Organic City tile and the Box Cities specifically address object reconstruction, i.e. tessellation and 3D building modeling. To obtain reference data like point clouds and DSMs for the underlying algorithms, the respective parts of the synthetic scene must undergo ray casting depending on the sensor configuration to be simulated. The output is then passed to the reconstruction software, either in its original form or artificially distorted to look more realistic. The reconstruction result like meshes can eventually be compared to the ground truth to derive error statistics. Quality criteria may include the distance of the regained vertices to the perfectly aligned polygons or the pixel-wise difference between the DSMs derived from the ground truth and the reconstruction.

5.6 Reverse engineering of unknown camera models

When oriented images prepared by closed-source photogrammetric software are to be processed using custom tools, the underlying camera model may not be exactly known. The lack of proper documentation particularly is critical for the correct interpretation of rotation angles, transformation sequences and distortion models. One approach to solving this problem is to render suitable image samples from the synthetic 3D scene together with a well-known orientation, for instance using POV-Ray. If the closed-source program can be convinced to process this input, the resulting camera model can be compared to the ground truth. The unknowns in the camera model may be investigated more systematically, and missing internal functions possibly can be disclosed in less time than with the brute-force method. In fact, to generate the respective orientation files, the outlined approach has already been successfully applied to validate the SDL code for the transformation between the camera models of POV-Ray and the original SGM implementation, and between POV-Ray and SURE (see section 4).

6. CONCLUSION

This paper has presented the design and composition of a synthetic scene for the validation and performance evaluation of photogrammetric algorithms. Unlike real-world data, the computer-generated 3D model specifically aims at software engineers in both early and late development stages where geometric inaccuracies due to measurement errors cannot be accepted. It has been outlined how images, point clouds and orientation samples can be derived from the three-dimensional ground truth with little programming effort. Validation recipes have been composed and followed for typical classes of photogrammetric algorithms, including the statistical assessment of the quality of texture mapping. Due to the exclusive use of open and human-readable data formats, the synthetic 3D scene and derived data products can be used and adapted as necessary to test and debug custom code. In the near future, it is planned to make the scene and the related software tools available to the public.

REFERENCES

Blinn, James F., 1982. A Generalization of Algebraic Surface Drawing. *ACM Trans. Graph.*, 1, 235–256. http://doi.acm.org/10.1145/357306.357310.

Frommholz, Dirk, Linkiewicz, Magdalena, Meißner, Henry, Dahlke, Dennis, 2017. Reconstructing buildings with discontinuities and roof overhangs from oblique aerial imagery. *ISPRS Hannover Workshop: HRIGI 17 - CMRT 17 - ISA 17 - EuroCOW 17*, XLII-1-W1, Copernicus Publications, 465–471.

Gerke, Markus, Rottensteiner, Franz, Wegner, Jan D, Gunho Sohn, 2014. ISPRS Semantic Labeling Contest. http://www2.isprs.org/commissions/comm2/wg4/potsdam-2d-semantic-labeling.html (31 January 2019).

Hirschmüller, Heiko, 2008. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30, 328–341. http://dx.doi.org/10.1109/TPAMI.2007.1166.

Huxhagen, U., Kern, F., Siegrist, B., 2009. Vorschlag für eine TLS-Prüfrichtlinie. *Photogrammetrie, Laserscanning, Optische 3D-Messtechnik.*, Wichmann, Heidelberg, 3–12.

IEEE Computer Society, 2008. *IEEE Standard for floating-point arithmetic*. IEEE, New York, USA.

Kelsey, Bart, 2015. OpenGameArt Artwork Repository homepage. http://opengameart.org (08 February 2015).

Klaus, Stephan, 2010. The solid trefoil knot as an algebraic surface. *CIM Bulletin*, 28, 2-4.

Knuth, Donald E., 1997. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Lorensen, William E., Cline, Harvey E., 1987. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *SIGGRAPH Comput. Graph.*, 21, 163–169. http://doi.acm.org/10.1145/37402.37422.

Nex, F.C., Gerke, M., Remondino, F., Przybilla, H.-J., Bäumker, M., Zurhorst, A., 2015. ISPRS benchmark for multi-platform photogrammetry. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3/W4, 135–142.

nFrames GmbH, 2019. SURE Knowledge Base. https://nframes.atlassian.net/wiki/spaces/SKB (1 April 2019).

OpenFootage, 2018. openfootage.net - HDRI panorama, Timelapse, Textures, 3D scan, smoke fire footage, panorama photography. https://www.openfootage.net (7 November 2018).

Perlin, Ken, 1985. An Image Synthesizer. *SIGGRAPH Computer Graphics*, 19, 287–296. http://doi.acm.org/10.1145/325165.325247.

Persistence of Vision Pty. Ltd., 2018. Persistence of Vision Raytracer 3.7. http://www.povray.org (30 January 2019).

Ramey, Diane, Rose, Linda, Tyerman, Lisa, 1995. MTL material format. *File Formats*. http://paulbourke.net/dataformats/mtl (26 March 2019).

Rothermel, Mathias, 2016. Development of a SGM-based multi-view reconstruction framework for aerial imagery. PhD thesis, University of Stuttgart.

Rottensteiner, Franz, Sohn, Gunho, Jung, Jaewook, Gerke, Markus, Baillard, Caroline, Bénitez, Sébastien, Breitkopf, U, 2012. The ISPRS benchmark on urban object classification and 3D building reconstruction. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3.

Scharstein, Daniel, Hirschmüller, Heiko, Kitajima, York, Krathwohl, Greg, Nešić, Nera, Wang, Xi, Westling, Porter, 2014. High-resolution stereo datasets with subpixel-accurate ground truth. *German Conference on Pattern Recognition Proceedings*, 8753, 31–42.

Segal, Mark, Atteley, Kurt, 2017. *The OpenGL Graphics System: A Specification*. 4.5 edn, The Khronos Group Inc.

The Khronos Vulkan Working Group, 2019. *Vulkan 1.1.105 - A Specification*. 1.1.105 edn, The Khronos Group Inc.