Multi-Perspective Detail+Overview Visualization for 3D Building Exploration

Sebastian Pasewaldt   Matthias Trapp   Jürgen Döllner

Hasso-Plattner-Institut, University of Potsdam, Germany
{sebastian.pasewaldt, matthias.trapp, juergen.doellner}@hpi.uni-potsdam.de

Abstract

Virtual 3D building models, as key elements of virtual 3D city models, are used in a growing number of application domains, such as geoanalysis, disaster management and architectural planning. Visualization systems for such building models often rely on perspective or orthogonal projections using a single viewpoint. Therefore, the exploration of a complete model requires a user to change the viewpoint multiple times and to memorize the content of each view to obtain a comprehensive mental model. Since this is usually a time-consuming task, which implies context switching, current visualization systems use multiple viewports to simultaneously depict an object from different perspectives. Our approach extends the idea of multiple viewports by combining two linked views for the interactive exploration of virtual 3D buildings model and their façades. In contrast to traditional approaches, we automatically generate a multi-perspective view that simultaneously depicts all façades of the building in one overview image. This facilitates the process of obtaining overviews and supports fast and direct navigation to various points-of-interest. We describe the concept and implementations of our Multiple-Center-of-Projection camera model for real-time multi-perspective image synthesis. Further, we provide insights into different interaction techniques for linked multi-perspective views and outline approaches of future work.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces (GUI) I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques I.3.3 [Computer Graphics]: Picture/Image Generation—Digitizing and scanning, Viewing algorithms

1 Introduction

Virtual 3D building models (V3DBMs) [GKNH12] represent core components of virtual 3D city models that are used in various application domains (e.g., environmental analysis, urban planning and disaster management). Typically, V3DBMs are explored and modified using interactive visualization and modeling tools, which often depict the models with a central or orthographic perspective projection in a single viewport. Due to the projection, the resulting image depicts the V3DBM only partially. To explore a complete building model (e.g., all façades or surfaces), a user needs to change the position and viewing direction of a virtual camera multiple times, and is required to memorize the content of the views to construct a comprehensive mental model of the V3DBM. Changing and memorizing
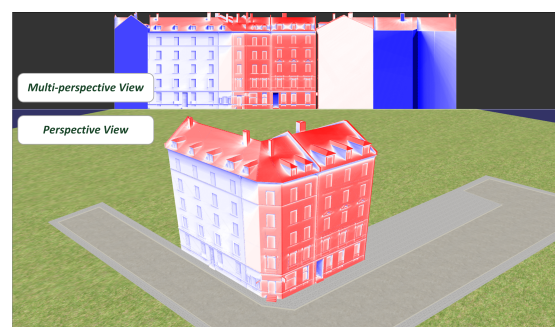

Figure 1: *Our detail+overview visualization combines a multi-perspective panoramic image (top) with an 3D perspective view (bottom). The overview enables direct navigation of the 3D view. The thematic color mapping represents the results of a solar potential analysis similar to [ED09].*

different views demands increased attentiveness of a user, which implies an increased cognitive workload. For example, Plumlee and Ware investigated, how the additional interaction using 3D zooming user interfaces influences the user's performance on comparison tasks. A key component in their model for the estimation of task completion time is the *"visual working memory capacity"* [PW06]. It denotes the capability of a user to temporally store – or remember – visual attributes, such as color and structure of geometric objects. Due to limitations of this capability, a user can store and recap only a limited amount of visual information [BKA11]. In a series of experiments, Plumlee et al. showed that with increasing complexity of a scene (e.g., number of geometric objects), zooming user interfaces become less effective for geoanalysis tasks. Instead they suggest to use multi-viewport visualization systems, which simultaneously show different views on the data in separated viewports.

In the context of V3DBMs, the multi-viewport systems are commonly used in modeling and CAD tools (e.g., Autodesk's 3D Studio Max and Maya). Here, usually four viewports depict a 3D model using four different views, which lowers interaction overhead and facilitates a precise estimation of 3D space [TKAM06]. Multi-viewport systems can also be used to depict a V3DBM at different scales or LOD for exploratory visualization [Rob07]. A distinct variant of these systems are detail+overview visualization tools [CKB09]. These tools show selected regions-of-interest at high detail and preserve a global overview at reduced detail. This principle basically uses *different views, windows, or viewports* to display both, details *and* overview. The overview can be an inset panel, a panel beside the detailed view, or another window in the case of a multi-window application.

In this paper we present a novel multi-viewport detail+overview visualization that combines a 2D multi-perspective view (overview) with a 3D perspective view of a V3DBM (detail view) (Figure 1). We focus on the exploratory visualization of surface or façade related information [LD10] exemplified by solar-potential analysis [ED09]. The key component of our tool is a *2D multi-perspective building panorama* used in the overview viewport that simultaneously and seamlessly shows all façades of the V3DBM in a single image and is synthesized in real-time. This reduces the number of viewports necessary to depict an overview of the complete model to a single viewport. By linking the detail and overview viewports, the building panorama enables a direct and simple navigation of the 3D detail view's virtual camera. Further, users who interact with the 3D detail view are supported by visual feedback in the overview, to locate their position in the 3D view in the context of the complete V3DBM. In general, a single multi-perspective building panorama is suitable for V3DBM of convex and non-convex shape that does not contain inner

yards. It is able to represent the surface of a 3D object that can be encircled (e.g., building blocks).

This paper contributes the following: (1) it presents an interactive *multiple-center-of-projection* (MCOP) rendering technique that generates multi-perspective images of virtual 3D building models and their façades based on the building geometry; (2) it introduces a novel detail+overview visualization concept that combines a 3D perspective view and a continuous 2D multi-perspective building panorama; (3) it presents interaction techniques that focus on an interactive exploration of V3DBMs in the context of exploratory visualization of thematic data.

2 Related Work

2.1 Multi-Viewport Visualization

Multi-viewport systems are commonly used in interactive 3D modeling software. Steinicke et al. adopted this concept for a 3D residential city planner that can be used to model V3DBMs based on 2D cadastral data [SRHB06]. Here, each viewport shows a different view on the V3DBM that is best suited for a partial modeling task. For example, the 2D cadastral view is well suited to modify the building footprint or select building façades, whereas the 3D view is especially suited to review modifications on the model. They evaluated their tool with domain experts and pointed out that different views support the user at specific modeling tasks and increases the accuracy.

The usage of multiple 2D viewports for façade-specific modeling operations is also used in the semi-automatic building reconstruction tools presented in [XFT$^+$08] and [WPK12]. Here, 2D painting and interaction metaphors are used to modify vector masks that represent prior detected geometric building features. Direct feedback of the modifications is given in a linked 3D viewport. Our visualization also uses multiple linked viewports for the exploration of building model façades, but in contrast to the above mentioned systems, it synthesize a multi-perspective view in real-time to depict the complete V3DBM instead of using multiple viewports with single views. Thus, it reduces the number of viewports to two: a 2D view for the overview and a 3D view for the detailed view.

2.2 Multi-perspective Views

Multiple camera models are capable to generate multi-perspective images. In general, these camera models can be categorized into *single* and *multiple* center-of-projection (SCOP and MCOP) camera models, which can utilize linear or non-linear viewing rays. A comprehensive review of various multi-perspective camera models is given in [YMS08]. For example, the RTcam (Rational-Tensor camera) of Hall et al. [HCS$^+$07] unifies SCOP and MCOP with linear or non-linear viewing rays, and produce multi-perspective images similar to our building panorama. The Graph Camera of [PRAV09] represents a MCOP camera model with
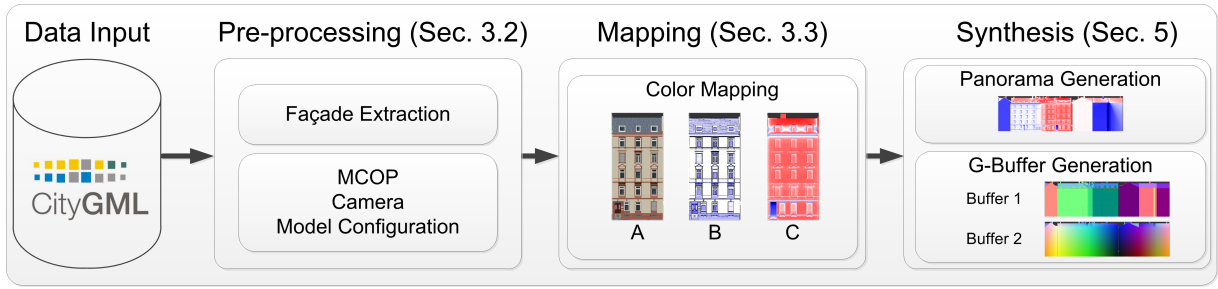
Figure 2: *A building is approximated by the building footprint (red outline). For each building wall, a virtual camera is placed to synthesize the respective façade image.*

(non-)linear viewing rays, which is also capable of depicting a V3DBM from multiple viewpoints at once. However, its design focuses on a wide field of applications. Agrawalla et al. [AAC+06] a rendering algorithm that takes multiple perspective input images to generate a multi-viewpoint road panorama. Compared to other approaches (e.g., [RGL04] and [RB98]), they focus on combining as large as possible parts of the input images for their final panorama. As a result their panorama features locally correct perspective views that offer depth cues, while the overall image exhibits less distortion.

We decided to utilize a multiple center-of-projection camera model, similar to the slit-camera of Rademacher and Bishop [RB98] or the cross-slit camera of Roman et al. [RGL04], because it is directly suitable to our purposes. The configuration of the cross-slit camera model can be automatically computed based on the V3DBM: multiple center-of-projections and the horizontal slits are defined by a camera path that is computed from the building footprint. The vertical slit is set to infinity to generate orthographic views.

3 Multi-perspective Panoramas

One of the main design goals for the multi-perspective building panorama is to provide overview to enable a user to easily generate a mental image of the V3DBM. Thus, it is important to depict the dominant elements of a building, i.e., the walls or façades, topological correct order. We use a MCOP camera model that is automatically configured based on the input 3D building geometry (e.g., a CityGML building model). The mapping of the surface properties (e.g., thematic information) on the building façade can be modified by a user according to the preferences and use-cases.

This section provides an overview of the building panorama generation pipeline (Figure 2) and presents details for each of the pipeline stages. Further, the synthesis and synchronization of detail+overview viewports are described.

3.1 Preliminaries & Conceptual Overview

A building panorama generates a distortion-free and topologically ordered presentation of all building façades. Technically, a façade F is a single planar face, for example a `WallSurface` of a Level-of-Detail 2 (LOD 2) CityGML

model [GKNH12]. Additional building wall features of LOD 2-4 models, such as, `BuildingInstallations` (e.g., rain pipes and stairs) or `Opening` (e.g., windows and doors), may be topological parts of F, but semantically and geometrically disjunct elements and are not considered during pre-processing.

The workflow of the panorama generation technique is described based on a CityGML representation of a V3DBM, i.e., a XML-based description of geometric, semantic and topological data. More generally, every mesh-based V3DBM with an additional vector-based footprint or camera path (e.g., following a road network or traversing along a housing block) can be processed by the pipeline. The surface-related thematic information are encoded in 2D raster images [LD10].

The pre-processing phase consists of two stages: (1) façade extraction and (2) MCOP camera model configuration. First, all building façades are extracted from the V3DBM and stored in an ordered list $\mathcal{F} = (F_0, ..., F_{n-1}), n \in \mathbb{N}$. Afterwards for each $F_i \in \mathcal{F}$, a virtual camera C_i is configured that approximates F_i, i.e., the camera's field-of-view covers F_i (Figure 3) completely.

Subsequently, in a mapping phase, a user can configure which surface properties are mapped on the building façades. Typical configurations include a photo-realistic texture (Figure 2A) or thematic surface properties (Figure 2C) [LD10]. In this example the thematic surface properties are encoded in a 2D raster image, where each pixel encodes a value $(v_{x,y})$ in the normalized value domain $[0, 1]$. The user can define a mapping of $v_{x,y}$ to a number of different color mappings. The mapping configuration can be independently configured for the overview and the 3D detail view.

In the last phase, the building panorama is synthesized based on the MCOP camera configuration ($\mathcal{C} = (C_0, ..., C_{n-1})$) and the user-defined color mapping. In addition to the visual representation, two G-Buffer [ST90] images are generated that are utilized for the viewport synchronization (Section 3.4), post-processing stylization (Section 5.2).
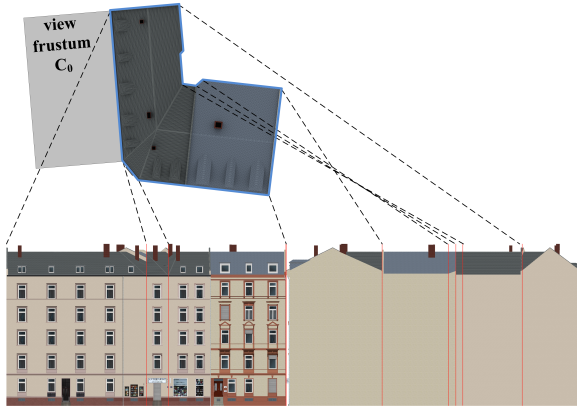
Figure 3: *A building is approximated by the building foot-print (red outline). For each building façade, a virtual camera is placed to synthesize the respective façade image.*

3.2 Façade Extraction & Camera Configuration

This section presents the algorithms for the façade extraction and the subsequent configuration of the MCOP camera model based on a CityGML building model. On model load, all `Wallsurface` ($\mathcal{W} = (W_0, \dots, W_{m-1})$) are extracted from the V3DBM. In addition the `GroundSurface` is extracted. The `GroundSurface` is represented as a `Polygon` (blue line in Figure 3), which is used to order $\mathcal{W}$ counter-clockwise. For each building wall $W_i \in \mathcal{W}$ of the ordered list, the planarity is checked. Non-planar building walls, i.e., walls that form a corner, are subdivided into planar subsections. The final list of topologically ordered and planar walls meet the definition of the building façades and form $\mathcal{F}; n \geq m$

Based on $\mathcal{F}$ the MCOP camera model is configured. For each $F_i \in \mathcal{F}, 0 \leq i < n$ a virtual camera configuration C_i is generated. To reduce distortion artefacts, a orthographic projection is used. This projection preserves angular and length relation and thus enables measurement operations. A single camera (C_i) of the MCOP camera model ($\mathcal{C}$) is defined and configured as follows: $C_i = (v_{to}, v_{from}, w, h, f_{near}, f_{far})$ (Figure 4):

- $v_{to} \in \mathbb{R}^3$ is the camera's look-to vector. It is placed at the geometric center of F_i.
- $v_{from} \in \mathbb{R}^3$ is the camera's look-from vector. It is computed by slightly translating v_{from} along the façade normal N_i.
- w is the view frustum's width. It directly correlates with the width of F_i.
- h is the view frustum's height. To ensure a consistent multi-perspective image and comparability between different façades, h is equal for all C_i and is based on the maximum height of all F_i.
- f_{near} is the distance of the near-clipping plane to v_{from}. It can be configured by the user to include or exclude oc-
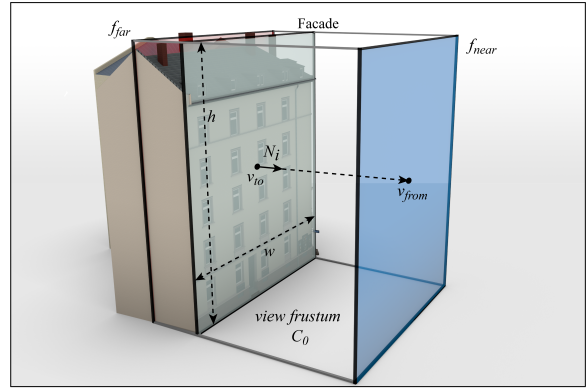

Figure 4: *Illustration of the camera configuration. The virtual camera is automatically configured to span the complete façade. The user can control the near and far-clipping plane to include or exclude further building objects. For example the far-clipping-plane is set to the roof's ridge line to include the roof and chimneys.*

cluding non-façade elements (e.g. vegetation or city furniture)

- f_{far} is the distance of the far-clipping plane to v_{from}. It can be configured by the user to include or exclude roof elements (e.g., chimneys) into the building panorama.

This configuration ensures, that each virtual camera C_i spans the corresponding building façade. Due to the topological ordering of $\mathcal{F}$, the list of all virtual cameras $\mathcal{C} = (C_0, \dots, C_{n-1})$ is ordered as well and thus generates a topological correct building panorama (Figure 3).

3.3 Viewport Configuration

The proposed visualization is composed of two viewports: the detail viewport depicts the V3DBM using a 3D perspective view, which enables interactive exploration using standard interaction techniques, such as zooming, panning and rotation. The overview viewport shows all façades of the V3DBM using the generated building panorama (Figure 1). The visible content of the overview can be modified using pan and zoom interaction techniques. The screen size and position of both viewports can be adjusted to the user's needs.

The initial view setup is configured in a way, that the 3D detail view depicts the V3DBM from a moderate distance, i.e., the view frustum encloses the complete building geometry. The overview is adjusted based on the image dimensions of the building panorama to maximize the occupied screen space. Both view configurations are saved as visual bookmarks, to enable a user to easily return to the initial view configuration.

During run time, a user can configure which surface properties are mapped to the 3D detail view and the building panorama. For example the user can map thematic data (e.g., solar radiation, noise pollution [MK10] or building

heat transmission [Dav04]) to one of a pre-defined color mappings for the 3D detail view and the building panorama (Figure 5) to facilitate exploratory visualization. If an interesting pattern has been identified, the mapping of the 3D detail view can be set to a photo-realistic mapping (e.g., depict the original façade textures), e.g., to investigate the origin of the pattern.

3.4 Viewport Synchronization and Visual Proxies

Both viewports independently depict the building using different perspectives at different levels-of-detail. This can complicate the mental mapping of a user's position in the 3D detail view and the corresponding position in the overview. However, this also offers new opportunities for an effective exploration of the V3DBM's façades. In the following we describe how visual proxies and viewport synchronization can be used to combine the benefits of the 2D overview and 3D detail view for an exploratory analysis of thematic data.

To support a user in locating its current position in the overview, we apply a 2D overlay on the overview as visual proxy that depicts the view frustum of the 3D detail view's virtual camera (Figure 5). The overlay is computed based on G-Buffer content [ST90], i.e., for each pixel of the building panorama, the corresponding 3D position and the surface normal vector are encoded in two additional images. Given these G-Buffer content and the current 3D view's projection- and view-matrix, one can compute for each pixel of the overview whether it is contained in the 3D view's frustum or not and apply a highlighting accordingly.

The camera configuration of the 3D detail view is synchronized on a per-frame basis resulting in a tight coupling of the 3D view and the panoramic overview. The frustum overlay enables a user to explore thematic information at a high zoom level in the detail view, while still locating the current position in the context of the complete V3DBM. Compared to a single 3D view the visual working memory can be relieved of memorizing the context, providing mental capacities for the analysis task.

4 Interaction Techniques

Besides using the building panorama to display additional context information, it mainly serves as an overview of all building façades. Thus, a user can identify points-of-interests as well as regions-of-interests (e.g., hotspots in the thematic data), and then use the 3D detail view to access more detailed information. To assist the user at this exploration task we link the 2D building panorama with the 3D detail view. Here, the building panorama serves as interaction proxy for a set of 2D interaction techniques such as point-and-click, hover, or sketch interaction, which synchronously modifies the virtual camera of the detail view.

Point-and-Click Interaction. To focus the 3D detail view to a point-of-interest, a user can click on the respective point in the building panorama. The virtual camera of the 3D de-
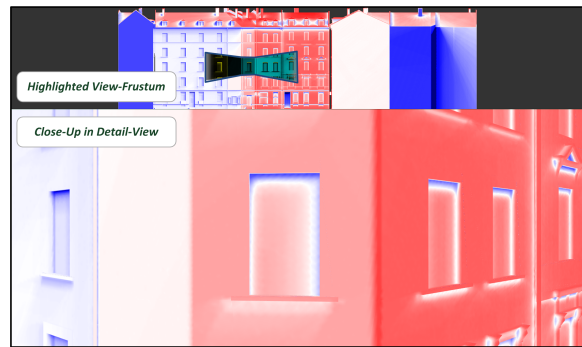

Figure 5: *By highlighting the view-frustum of the 3D view in the panoramic image the user can inspect details of the V3DBM, while still maintaining its position in the context of the complete building.*

tail view is then centred to the user-specified position using a viewing direction orthogonal to the building façade. This reduces perspective distortion and eases estimation of spatial relations such as areas, lengths, and angles. Using such direct navigation metaphors reduces the interaction overhead, since the user only has to perform a single click instead of applying a series of pan, rotation, and zooming interactions to navigate to a desired location.

HoverCam Interaction. Using image-based scene representations [ST90], we can also emulate established continuous 3D interaction techniques such as HoverCam [KKS⁺05]. During interaction, HoverCam ensures that a virtual camera hovers equidistant to, and always facing the 3D object. Thus, it eases the exploration of 3D objects and avoids "getting-lost situations" [BBD05]. Using the Hover-Cam interaction mode, mouse hovering in the overview results in a continuous and precise hovering of the virtual camera in the detail view, enabling continuous movement over all building façades.

Sketch Interaction. In addition to the click and hovering metaphor, we further provide sketching metaphors to modify the virtual camera of the 3D detail view or trigger measurement operations on the geometry. By sketching line segments in the overview the user can specify a camera path that results in a continuous movement of the details view's camera. Compared to the HoverCam interaction mode, the interaction is less precise and delayed, since the sketched 3D positions are interpolated by the camera path and the animation starts slightly after the user has finished sketching. By delaying the camera update, a user can first define the camera path and afterwards completely focus on the information presented in the detail view. The same metaphor is used for a measurement mode to compute 3D distances. To measure areas the user can sketch a rectangle, which is also displayed as a overlay in the overview. This metaphor is also used to adjust the 3D view to regions-of-interests.

Table 1: *Comparative performance evaluation for different building panorama configurations in milliseconds. The total time includes the time required for image-synthesis, post-processing and texture read-back.*

Model	#Vertices	#Cameras	#Pixels	Image synthesis	Stylization	Texture read-back	Total
Model 1	22,731	11	5,575,284	1.45	405.75	262.75	669.95
Model 1	22,731	11	1,072,500	1.42	18.45	48.75	68.65
Model 1	22,731	5	5,575,284	0.71	464.33	334.36	799.40
Model 1	22,731	5	1,072,500	0.73	18.07	46.96	65.76
Model 2	1,098,119	49	4,601,952	10.16	194.47	186.15	380.62
Model 2	1,098,119	49	1,072,512	9.63	34.29	43.47	87.39
Model 2	1,098,119	20	4,601,952	4.00	181.89	164.39	350.28
Model 2	1,098,119	20	1,072,512	4.26	25.05	37.50	62.55

5 Interactive Rendering of Multi-perspective Building Panoramas

The presented system is implemented in C++ using Open-SceneGraph (OSG) as scene-graph API and OpenGL wrapper, Qt 4.8 as Graphical User Interface (GUI) framework, OpenGL Shading Language (GLSL) for shader programming and Compute Unified Device Architecture (CUDA) for the image-based stylization. The import of CityGML V3DBM is implemented using libcitygml. The multi-perspective image synthesis is implemented using multi-pass render-to-texture (RTT) [PF05]. In addition to the color texture additional G-Buffer textures are generated.

5.1 G-Buffer Configuration

In addition to the RGB color texture, geometric and semantic information are encoded in two 32 bit RGBA textures. The vertex's world position ($v_{world} \in \mathbb{R}^3$) and the unique identifier ($id \in \mathbb{N}$) of the current camera C_i is stored in one texture. The vertex's normal ($n_{world} \in \mathbb{R}^3$) and depth in camera coordinates ($d \in \mathbb{R}$) are stored using an additional texture. During image-synthesis, the information is written to the textures, using multiple-render-targets.

The size of the textures are initialized based on the aspect-ratio of the building panorama and a user-defined maximum texture size. The aspect-ratio is computed based on the overall viewport of the MCOP camera model, i.e., the sum of all viewports sizes (width and height). The longest side of the overall viewport is mapped to the user-defined maximum texture size and the shorter side is computed according to the aspect-ratio.

5.2 Compositing & Stylization

The image synthesis of the multi-perspective building panorama uses multiple render-passes to render the successive C_i into one shared texture (Algorithm 1). For each rendering pass the vertex and fragment shaders are configured based on C_i, i.e., the view and projection matrix are adjusted. Further, a gradient texture ($G_{id} \in \mathcal{G}$), which is used to map the normalized thematic information to a user-defined color scheme, and the current camera id is passed to the fragment shader. To ensure that all $C_i \in \mathcal{C}$ are correctly rendered into the shared texture, the viewport is translated by the width of the prior rendered camera viewport.

Afterwards, the current view is rasterized to the G-Buffer. Optionally image-based rendering techniques, such as XDoG [WKO12] or edge enhancement, to include depth cues in the building panorama, which would be otherwise absent due to the equidistant orthographic camera setup.

5.3 Performance Discussion

Performance measurements were performed on an Intel Xeon CPU (Quad-Core, 2.66 GHz) with 6 GB RAM and a Nvidia GeForce GTX 580 (2 GB VRAM). The results (Table 1) indicate that the panorama image synthesis can be performed in real-time.

The time for image synthesis scales linear with the number of cameras and even for more complex models the computation can be performed in real-time. The bottleneck is the texture read-back between GPU and CPU as well as the post-processing pass. Both scale with the image resolution. For larger images (e.g., 5 Mio. pixels) the overall computation time exceeds interactive frame rates. To cope with this, post-processing as well as texture read-back are postponed. The post-processing is applied on the overview viewport on a per frame basis. Since this viewport has a significant lower resolution, the computation time decreases in our setup by factor 10. Since the two views of the prototype share a single OpenGL context, the texture read-back can also be post-

---

Algorithm 1 Building panorama image synthesis

1: *GBuffer ← setup*() [setup G-Buffers]
2: **for all** $C_i \in \mathcal{C}$ **do**
3:      $M_{view} \leftarrow M_{view_i} := fetch(i, \mathcal{C})$ [fetch view-matrix]
4:      $M_{proj} \leftarrow M_{proj_i} := fetch(i, \mathcal{C})$ [fetch projection-matrix]
5:      *setupVertexShader*(M_{view}, M_{proj})
6:      $G \leftarrow G_{id} := fetchGradientTexture(id, \mathcal{G})$ [fetch gradient]
7:      *setupFragmentShader*(i, G)
8:      $V \leftarrow F_i := fetch(i, \mathcal{C})$ [fetch viewport size]
9:      *setupViewport*(V) [Setup/Translate viewport]
10:      *rasterize*(C_i) [rasterize to G-Buffer]
11: **end for**
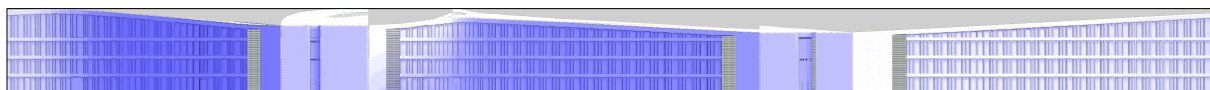12: *GBuffer ← postProcessingPass*(*GBuffer*)

Figure 6: *Example of a building panorama with a large horizontal extent. Due to the aspect ratio of 14:1 it occupies only a small part of the overall screen space and thus, only limited information can be communicated.*

poned to either the interaction process or the storage of the building panorama to secondary storage (e.g., harddisk).

Since the current image-synthesis does not support tile-based rendering, the prototype is graphics memory bounded. The additional amount of graphics memory O required for the G-Buffer images can be estimated by: $O = 3 \cdot w \cdot h \cdot 4 \cdot p$ bytes. Our prototype uses a precision $p = 4$ byte per channel, to ensure position accuracy for the interaction techniques. Further, the CUDA post-processing stage requires additional temporal textures.

6 Results and Discussion

This section briefly describes the application of the visualization concept for exploratory visualization of thematic data. Further it discusses limitations of the rendering technique and approaches for future work.

Linking the overview and the detail view offers a fast and direct way to explore the V3DBM's façades and its related thematic information. At the beginning of the exploration, the overview enables a user to (1) identify points- and regions-of-interests and then (2) navigate to these by using click and hover interaction. Afterwards, the information can be explored in more detail by adjusting the zoom-level of the 3D detail view. Providing the information about the view frustum in the overview, the detailed information can be analysed in the context of the complete building. Both, direct navigation metaphors and overview reduce the mental workload of a user and frees visual working memory capacities.

6.1 Limitations

The V3DBM (Figure 1-5) , which has been used as an example in this paper, is well suited for the panoramic overview since the ratio between its horizontal and vertical extent is small. Buildings with large vertical extent (e.g., skyscrapers) or with a large horizontal extend (Figure 6) result in a panoramic image that is either extremely tall or wide. To display these panoramas on a limited viewport size, the panoramic image has to be scaled drastically, thus less information can be communicated. One approach to this challenge is to subdivided and re-arrange the panoramic image or to ask the user to specify regions, which are of respective interest.

Compared to other MCOP camera model configurations, our configuration does not capture all details of the V3DBM and can include geometric features multiple times. In our example small protrusions, such as shop signs or the staircase, are not included in the image and the chimneys are depicted multiple times. However, for our use case this can be ne-

glected because the requirement on the building panorama is not being complete but rather to provide a topological correct and homogeneous overview.

6.2 Approaches for Future Work

We are currently working on a user evaluation of the presented concept for exploratory visualization of V3DBM to identify further improvements, such as a different MCOP camera model configurations or alternative interaction techniques. In addition, we plan to investigate the applicability of the concept for other domains, such as communication of architectural design decisions or building refactoring.

The MCOP camera model configuration is currently based on the building's façades and footprint. As a consequence, it generalizes from the original building geometry (e.g., omitting minor geometric features). We like to develop this idea further by applying generalization techniques for V3DBMs as proposed by Kada [Kad07]. The generalized variant is used as input for the MCOP camera model configuration while the image synthesis operates on the original variant. The resulting building panorama could be used as input for projective texture-mapping for the V3DBM of a lower LOD. The same approach could be used for generalized virtual 3D city models [GTD11].

Further, we would like to implement a tile-based image synthesis and post-processing to overcome the VRAM limitation and support the generation of Gigapixel images. Furthermore, the 2D multi-perspective G-Buffer images represent a mapping of the 3D model space to 2D image space. This enable to shift computational analysis task from 3D to 2D, which is a promising approach to increase the performance of geo-analysis tasks, e.g., by applying 2D high-performance GIS algorithms.

7 Conclusions

This paper presents concepts and implementation of multi-perspective building panoramas and their integration into a novel detail+overview visualization. The detail+overview visualization and the corresponding interaction techniques aim for assisting the user at the interactive exploration of surface-related properties (e.g., thematic information) of virtual 3D building models. The visualization concept is based on linking a 2D multi-perspective overview, containing a panoramic image of all building façades, with a 3D perspective detail view. The building panorama facilitates an overview and the identification of hotspots in the thematic data. The presented interaction metaphors support the direct and fast exploration of virtual 3D building models while still

locating one's own position in the context of the complete virtual building using visual proxies.

The presented concept for an automatic generation of the multi-perspective building panorama can be easily extended to depict objects of a larger cartographic scale (e.g., building blocks) or to depict only subsets of the building model (e.g., a specific storey). These extensions facilitate new applications, such as generalized 2D depictions of virtual 3D building models or enable projective texturing for generalized virtual 3D city models. Further, the real-time image synthesis allows using the building panorama in modeling tools as an overview. In the future, we plan to investigate these possibilities as part of a user study to extend our concept and implementation.

Acknowledgments

References

[AAC⁺06] Aseem Agarwala, Maneesh Agrawala, Michael Cohen, David Salesin, and Richard Szeliski. Photographing long scenes with multi-viewpoint panoramas. In *ACM SIGGRAPH '06*, SIGGRAPH '06, pages 853–861. ACM, 2006. 3

[BBD05] Henrik Buchholz, Johannes Bohnet, and Jürgen Döllner. Smart and Physically-Based Navigation in 3D Geovirtual Environments. In *Proc. IEEE IV*, pages 629–635. IEEE Computer Society Press, 2005. 5

[BKA11] Timothy Brady, Talia Konkle, and George Alvarez. A review of visual memory capacity: Beyond individual items and toward structured representations. *Journal of Vision*, 11(5):1–34, 2011. 2

[CKB09] Andy Cockburn, Amy Karlson, and Benjamin B. Bederson. A Review of Overview+Detail, Zooming, and Focus+Context interfaces. *ACM Comput. Surv.*, 41(1):2:1–2:31, 2009. 2

[Dav04] Morris Grenfell Davies. *Building heat transfer*. Wiley, 2004. 5

[ED09] Juri Engel and Jürgen Döllner. Approaches Towards Visual 3D Analysis for Digital Landscapes and Its Applications. *Digital Landscape Architecture Proceedings 2009*, pages 33–41, 2009. 1, 2

[GKNH12] Gerhard Gröger, Thomas Kolbe, Claus Nagel, and Karl-Heinz Häfele. OpenGIS City Geography Markup Language (CityGML) Encoding Standard Version 2.0.0. Technical report, Open Geospatial Consortium Inc., 2012. 1, 3

[GTD11] Tassilo Glander, Matthias Trapp, and Jürgen Döllner. Concepts for Automatic Generalization of Virtual 3D Landscape Models. In *Digital Landscape Architecture Proceedings 2011*, pages 127–135, 2011. 7

[HCS⁺07] Peter M Hall, John Collomosse, Yi-Zhe Song, Peiyi Shen, and Chuan Li. RTcams: A New Perspective on Nonphotorealistic Rendering from Photographs. *IEEE Trans. Vis. Graph.*, 13(5):966–979, 2007. 2

[Kad07] Martin Kada. Scale-dependent simplification of 3D building models based on cell decomposition and primitive instancing. In *Spatial Information Theory*, pages 222–237. Springer, 2007. 7

[KKS⁺05] Azam Khan, Ben Komalo, Jos Stam, George Fitzmaurice, and Gordon Kurtenbach. HoverCam: interactive 3D navigation for proximal object inspection. In *Proc. I3D*, pages 73–80. ACM, 2005. 5

[LD10] Haik Lorenz and Jürgen Döllner. 3D feature surface properties and their application in geovisualization. *Computers, Environment and Urban Systems*, 34(6):476–483, 2010. 2, 3

[MK10] Enda Murphy and Eoin King. Strategic environmental noise mapping: methodological issues concerning the implementation of the EU Environmental Noise Directive and their policy implications. *Environment International*, 36(3):290–298, 2010. 4

[PF05] Matt Pharr and Randima Fernando. *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*. Addison-Wesley Professional, 2005. 6

[PRAV09] Voicu Popescu, Paul Rosen, and Nicoletta Adamo-Villani. The graph camera. *ACM Trans. Graph.*, 28(5):1, 2009. 2

[PW06] Matthew Plumlee and Colin Ware. Zooming versus multiple window interfaces: Cognitive costs of visual comparisons. *ACM Trans. Comput.-Hum. Interact.*, 13(2):179–209, 2006. 2

[RB98] Paul Rademacher and Gary Bishop. Multiple-center-of-projection images. In *ACM SIGGRAPH'98*, pages 199–206, 1998. 3

[RGL04] Augusto Roman, Gaurav Garg, and Marc Levoy. Interactive Design of Multi-Perspective Images for Visualizing Urban Landscapes. In *Proceedings of the conference on Visualization '04*, VIS '04, pages 537–544. IEEE Computer Society, 2004. 3

[Rob07] Jonathan C Roberts. State of the Art: Coordinated & Multiple Views in Exploratory Visualization. In *Proc. IEEE CMV*, CMV '07, pages 61–71. IEEE Computer Society, 2007. 2

[SRHB06] Frank Steinicke, Timo Ropinski, Klaus Hinrichs, and Gerd Bruder. A multiple view system for modeling building entities. In *Proc. Coordinated and Multiple Views in Exploratory Visualization*, pages 69–78, 2006. 2

[ST90] Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-D shapes. *ACM SIGGRAPH'90*, 24(4):197–206, September 1990. 3, 5

[TKAM06] Melanie Tory, Arthur E Kirkpatrick, M Stella Atkins, and Torsten Möller. Visualization task performance with 2D, 3D, and combination displays. *IEEE Trans. Vis. Graph.*, 12:2006, 2006. 2

[WKO12] Holger Winnemöller, Jan-Eric Kyprianidis, and Sven Olsen. XDoG: an extended difference-of-Gaussians compendium including advanced image stylization. *Computers & Graphics*, 36(6):740–753, 2012. 6

[WPK12] Fabian Wanner, Fabrizio Pece, and Jan Kautz. Simplified User Interface for Architectural Reconstruction. In *TPCG*, pages 37–40. Eurographics Association, 2012. 2

[XFT⁺08] Jianxiong Xiao, Tian Fang, Ping Tan, Peng Zhao, Eyal Ofek, and Long Quan. Image-based façade modeling. *ACM Trans. Graph.*, 27(5):161:1–161:10, 2008. 2

[YMS08] Jingyi Yu, Leonard McMillan, and Peter Sturm. Muliperspective Modeling, Rendering, and Imaging. *Comput. Graph. Forum*, 29:227–246, 2008. 2