Computers & Graphics 75 (2018) 25–35

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

# Fast cutaway visualization of sub-terrain tubular networks

Artem Konev[a,*], Manuel Matusich[b], Ivan Viola[b], Hendrik Schulze[a], Daniel Cornel[a], Jürgen Waser[a]

[a] VRVis Forschungs-GmbH, Donau-City-Straße 11, Wien 1220, Austria
[b] TU Wien, Favoritenstraße 9–11, Wien 1040, Austria

ARTICLE INFO

*Article history:*
Received 26 March 2018
Revised 11 July 2018
Accepted 11 July 2018
Available online 17 July 2018

*Keywords:*
Cutaway visualization
Procedural billboarding
Subsurface networks

ABSTRACT

This paper proposes a context-preserving 3D visualization technique for interactive view- and distance-dependent cutaway visualization that reveals the subsurface urban infrastructure network. The infrastructure itself is displayed using the procedural billboarding technique, and its internals are revealed through a new cutaway algorithm that operates directly on the procedurally generated structures in the billboard proxy geometry. Both described cutaway techniques achieve interactive frame rates for the infrastructural network of a mid-sized city. Performance benchmarks and a domain expert evaluation support the potential usefulness of this technique in general and its particular utilization for the sewer network visualization.



## 1. Introduction

Illustrations are often used to convey complex internal characteristics of a three-dimensional structure. They are employed in wide-ranging domains including medicine, mechanical engineering, architecture, and art among others [1]. To understand the internal complexity of such entities, the viewer needs to look inside them [2]. However, most of the structural elements are occluded by each other and the enclosing surface. It is therefore important to reduce the mutual visual interference of the occluding elements and let the viewer focus on the element of interest.

To do so, many illustrators make the outer structures transparent in order to reveal the otherwise hidden internal details. Such strategy allows, to a certain degree, to perceive simultaneously the outer geometry and the internal details. However, when shading multiple surfaces enclosing one another, the perception of the shading might result in several interpretations of the observed geometry. As an alternative to transparency-based techniques, cutaway-based approaches enable a clear, unoccluded view of the internal structures. They utilize the outer geometry as a context, thereby supporting the depiction of the internal details without any ambiguity. Parts of the outer layers are removed from the visualization by means of a cutaway geometry.

For 3D data visualization, multiple interactive cutaway techniques have been inspired by hand-crafted illustrations, especially in the context of medical, technical, or geological data visualization. In medical visualization, cutaways reveal compact internal structures, for which algorithms have been designed. One traditional application of cutaways in medicine is for educational illustrations of human anatomy. A more recent example, as Burns et al. suggest [3], is the the display of real-time imaging data (e.g., ultrasound) embedded into dense 3D Computed Tomography scans to assist medical interventions. In the technical domain, cutaways are widely used to illustrate the internal structure of complex mechanisms [4] or processes taking place inside those mechanisms. Geological illustrations often utilize cutaway techniques for revealing important subsurface details [5]. In visualization of geological models, structures can be compact (e.g., an oil trap), or more spread out (e.g., a network of channels originating from ancient fluvial systems). The geological cutaway techniques typically do not present a complex and detailed scenery on the Earth surface itself, these only provide a sense of orientation.

In urban environments, cutaways can help to expressively communicate subsurface infrastructure of a subway line [6], as illustrated in Fig. 1. For urban subsurface infrastructure, it is essential to communicate surface characteristics such as streets, buildings, parks, and water systems as there is a very strong relationship between them and the underlying subsurface infrastructure. It is usually not desirable to compromise on the visualization of the surface features in favor of the underlying features. A cutaway should rather be limited to structures that are very important or are of the viewer's interest and to their immediate neighboring structures. Straightforward application of existing cutaway visualization algorithms tailored for other spatial data distribution would result in cutting away the entire urban surface level. Therefore,

* Corresponding author.
*E-mail address:* konev@vrvis.at (A. Konev).

https://doi.org/10.1016/j.cag.2018.07.004

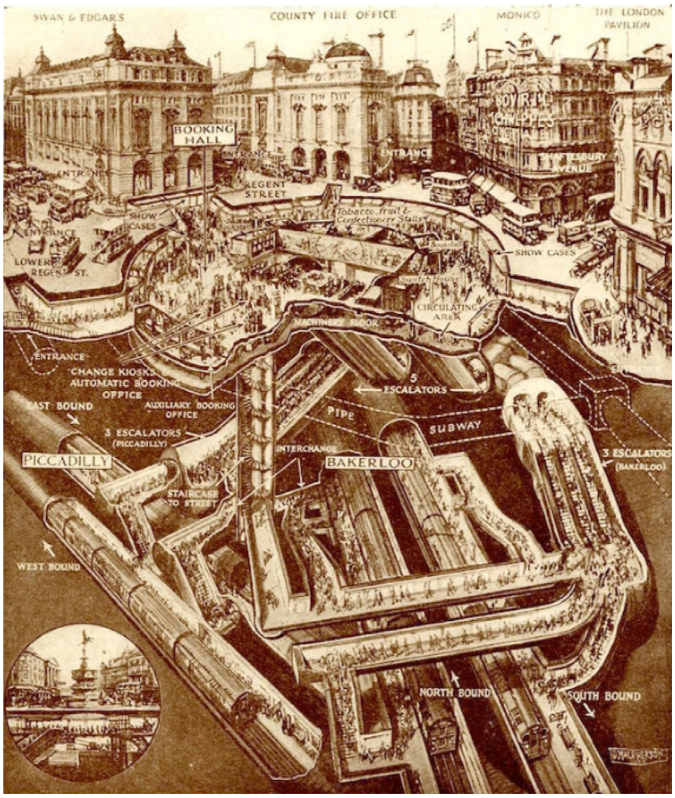


**Fig. 1.** Cutaway illustration of the Piccadilly Circus underground station to explain the complexity of the construction works, 1930. (c) London Transport Museum Collection.

another controlling mechanism for cutaways is needed in addition to the viewing direction. Our system incorporates a new parameter – distance to the surface – that defines whether a cutaway structure should be formed to reveal the subsurface detail or surface-level details should be conveyed instead. A novel distance-based cutaway parametrization forms the first technical contribution of this paper.

Subsurface network infrastructure of a mid-sized city forms a large model, and its high geometric resolution will have significant impact on the rendering performance. To ensure interactive rendering frame rates for city-scale visualization, *lightweight* geometric representations are necessary. Urban infrastructure is geometrically defined through a simple geo-referenced node-link diagram. Instead of rendering complex geometry of the subsurface infrastructure, we utilize the concept of procedural billboards, where the geometric detail is analytically computed during the rendering stage. Thanks to this approach, the overall geometric complexity of the subsurface network is represented by the essential topology, and the geometric detail is computed on-the-fly just in case it becomes potentially visible. Using this procedural billboarding (aka impostor rendering) for representing geometry, however, raises the research question on how to realize cutaway visualization on this type of data. Cutaway visualization applied to procedural billboards forms the second technical contribution of this paper.

Our prototypical visualization system has been tested and demonstrated on a city-scale sewer network infrastructure for the German city of Cologne (Fig. 2). The input data contained polygonal meshes in the .OBJ format to represent terrain, buildings, and vegetation. The sewer data was provided as a set of lines in the .SHP format (standard ESRI shape files). Additionally, water level information was attributed to each sewer pipe and sewer shaft. The use of our technique reduces the frame rate insignificantly, making the interactive visualization of the entire city scale possible on consumer graphics hardware. The visual encoding of the new cutaway techniques has been discussed with architectural visualization designers and with domain representatives who study subsurface sewer networks in the context of city flood management.

## 2. Related work

Seligmann and Feiner introduced the first algorithm for computer-generated cutaways [7]. Already in their work the cutaway geometry exhibited the characteristic zig-zag pattern, indicating the artificiality of the cut [8]. Diepstraten et al. [9] made an attempt to extract a set of rules from traditional cutaway techniques. These rules were further used to evaluate and compare methods of cutaway drawings. Although, according to the authors, the formalization of such rules was a hard task, five convincing rules for fully automatic cutaway drawings were provided. They are as follows: (1) *Inside and outside objects have to be distinguished from each other*, (2) *The cutout geometry is represented by the intersection of (a few) half spaces*, (3) *The cutout is located at or around the main axis of the outside object*, (4) *An optional jittering mechanism is useful to allow for rough cutouts*, and (5) *A possibility to make the wall (cut surface) visible is needed*. We make use of these rules to assess our approach. Another commonly used cutaway technique is the Stencil Test. Thereby convex exterior objects can be cut open. Our image-space technique uses the stencil test and stencil buffer to represent the cutout geometry.

In the importance-driven volume rendering technique presented by Viola et al. [10], the importance values are assigned to the data parts. An additional property encoding the visibility priority is then assigned to each voxel of the volumetric data set. In traditional volume rendering, a transfer function based on density is used to compose a final image. With additional information of importance and various composing schemes using this new dimension in the rendering pipeline, the technique of Viola et al. makes the object of interest clearly visible. In case a less important object occludes more important structures it is displayed more sparsely in the obscuring area. Bruckner and Gröller [11] present VolumeShop, an interactive system for the semi-automatic generation of illustrations. Their approach targets volumetric data and makes use of a dynamic combination of cutaway and ghosting techniques. Ropinski et al. [12] introduce interactive lenses that allow the user to define interesting regions of volumetric datasets. Different rendering techniques such as transparency or cutaways are then automatically applied to these regions. Bruckner et al. [13] propose to only suppress those regions of volumetric data that do not contain prominent features. To achieve this, the authors compute the opacity of each sample as a function of shading intensity, gradient magnitude, distance to the eye point, and the previously accumulated opacity. This approach preserves important context information such as shape cues. Krüger et al. [14] present a similar approach featuring interactive, continuous blending between different surface representations in a volume, resulting in an intuitive, depth-preserving highlighting.

Arguably, the most prominent application domain where cutaways and ghosting techniques have been demonstrated is medicine. Straka et al. used cutaways and ghosted views for blood vessel visualization [15] to provide a clear view on the vessel lumen while conveying contextual skeletal structures. Krüger et al. apply cutaways for the neck dissection planning, where lymph nodes are selectively shown using cutaways and ghosted views [16]. Another notable application is tumor surgery planning from abdominal organs, an anatomical area where many structures obstruct the view between pathological structures and the viewpoint [17]. Anatomical scans can be also used in the

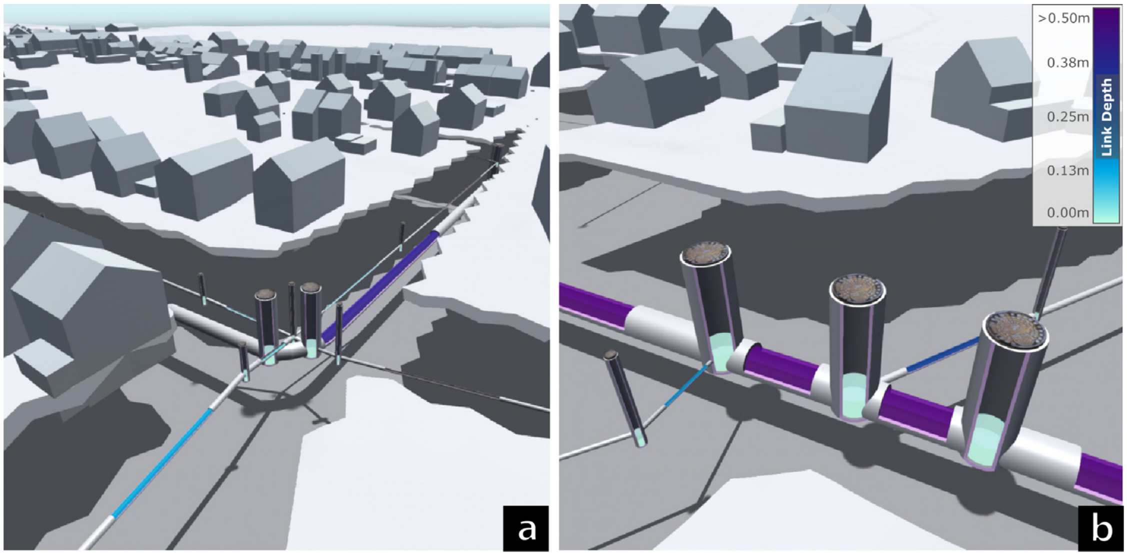


**Fig. 2.** Context-preserving visualization of an underground sewer network. (a) The terrain above the sewer elements is automatically cut open to display the occluded pipes and shafts. (b) The cutaways on the surface of the pipes (rendered as billboards) and shafts reveal the water depth inside the sewer system. The color of the water surface in the pipes gives an additional indication of the water depth.

context of medical education. Here, Mensmann et al. [18] have presented a technique for opening up the contextual tissue to reveal the structures beneath. Birkeland and Viola [19] have proposed view-dependent peel-away views for similar purpose. Wang et al. [20] proposed to combine magic lens metaphor with open-up views using the ray-deflection strategy.

Ropinski et al. [21] propose a method for Virtual Reality-based exploration of seismic datasets. By using VR-based interactions, the user can specify regions of interest and apply to them different clipping and transparency-based visual representations to avoid occlusion. Patel et al. [22] have presented an algorithm to generate textbook like seismic illustrations. A number of layers are combined such as 2D textures and 3D volumes, and the 3D volumes are blended with the 2D textures by 1D transfer functions. Beside the applications in medicine, to support the exploration of natural oil reservoir models, De Carvalho et al. [23] suggest an approach for an interactive cutaway generation based on screen-space GPU techniques. The approach is tailored for corner-point grids, a volumetric representation standard for the Oil and Gas industry. Given a set of “primary” volumetric cells to focus on, the algorithm of De Carvalho et al. extracts the camera-facing frustums from their bounding boxes. The union of these frustums is then rendered into a texture, against which a depth test is performed later on in the fragment shader. This approach allows for an on-the-fly generation of the cutaway volume and for cheap dynamic modifications thereof.

For polygonal meshes, Coffin and Höllerer [24] provide an rapid cutaway interface for modeling the cut explicitly by the user. The RibbonView [25] is similar to free-form-deformation embedding which is used for cutting away the occluding geometry, either entirely or leaving a small set of ribbons, indicating the original contextual structure. Interactively generated cutaways of complex three-dimensional models are discussed by Li et al. [26]. That paper includes a comprehensive overview of cutting conventions found in traditional scientific and technical illustrations. The key difference to previously published works is that the basis for the cutaway shape is not extracted from the geometry of the occludee, but from the geometry of the occluder instead. The first method that achieved interactive frame rates with adaptive cutaways for comprehensible rendering of dynamic polygonal scenes was presented by Burns and Finkelstein [27]. For each frame, the cutaway is adapted to the point of view, and hence it is view-dependent in the same way as in the importance-driven technique [10]. Burns and Finkelstein also used an image-based technique, and the computation is able to handle complex scenes and models. Le Muzic et al. [28] propose a novel method for rendering cutaway illustrations of mesoscopic biological models. Such models are characterized by a large number of instances of very few types. In the approach of Le Muzic et al., the user can interactively control the visibility distribution of each type. The authors also introduce a so-called aperture effect for fuzzy removal of occluding instances. In this effect, a 2D distance transform is computed and used as a mask stored in a texture. In the fragment shader of occluding instances, the distance field is looked up and fragments are discarded according to a user-defined parameter. In the context of terrain cutaways, the design principles for cutaway visualization of geological models presented by Lidal et al. [29] are of interest.

Billboarding originally emerged as a technique to increase frame rates of real-time 3D graphics systems. In earlier stages, graphics systems and especially computer games struggled to improve the visual quality without losing frame rates. The actual term was originally used to describe a part of the technique behind dynamically generated impostors [30]. Alongside with other frame rate improving techniques like culling of non-visible objects or using several levels of detail, the use of impostors proved to be useful in that regard. The idea behind dynamically generated impostors was to replace distant polygon rendered objects with textures. These textures display the replaced object, that has previously been rendered into a texture buffer. Since the viewing angle on distant objects changes relatively minor, it is practically impossible to see any difference. Reusing the same texture from the buffer many times for similar objects can save a lot of computation and therefore allows to render more complex scenes. For an efficient visualization of particle trajectories, Schirski et al. proposed a method based on billboarding [31]. The so-called Virtual Tubelets use cylindrical billboards to simulate movement of particles through the flow domain. In essence, the cylindrical billboards are quadrilaterals that are always aligned toward the viewer. To finally get the illusion of self-illuminated tubes, the Virtual Tubelets are textured appropriately. In addition to the illumination, coloring is applied to visualize, for instance, the velocity of the particles on top of the respective trajectories. A recent example of modern use of impostors can be found in the work of Le Muzic et al. [32]. The authors introduce cellVIEW, a tool for illustrative and multi-scale rendering of large biomolecular datasets. Atoms of a cell are visualized as spheres. Since thousands of them have to be rendered, sphere meshes would result in low frame rates. Therefore, individual atoms of a cell are rendered via 2D depth impostors due to a much lower vertex number.

In the geospatial domain, some 3D geographic information systems (3D GIS) offer the possibility to display subsurface infrastructure. Most of them use the terrain transparency effect to remove occlusion [33]. Cornel et al. [34] use the combination of cutaway and transparency to display occluded building cellars. They also cut away building meshes to ensure the visibility of a certain, user-selected building. However, to our knowledge, this work is the first that proposed distance-controlled interactive cutaways and cutaway visualization of procedural billboards.

In Augmented Reality applications, information stored digitally is graphically overlaid on views of the real world. If the overlaid information represents 3D features occluded by real-world surfaces (terrain, walls, etc.), there is a clear use case for cutaway visualization [35]. Zhang et al. [36] show underground pipes in an AR application. To achieve the immersive experience, they render a pit in the terrain, where the pipes are shown. In a similar application, Li et al. [37] carefully treat the occlusion of the rendered pipes by the pit walls and real-world objects. The billboard-based techniques presented in our paper can be especially useful for such purposes, as the actual rendering happens on mobile devices and therefore can become a burden in case of a large-scale infrastructure. In the medical domain, Bichlmeier et al. [38] present a method for the in-situ visualization of 3D medical imaging data in the context of the patient's own anatomy recorded in real time. Transparency of the video image is locally adjusted to create an impression of a cut-away.

## 3. Overview of the approach

This section gives a higher-level description of the presented approach as applied to the sewer network infrastructure of the city of Cologne (Germany). The overview of the pipeline with its main stages is shown in Fig. 3.

In the first stage, terrain cutaways are made using a novel distance-based parametrization. For this purpose, a grayscale texture is created from the pipe lines, encoding the 2D distance to the closest pipe. In the fragment shader, based on a lookup in this texture and the camera distance, a decision is made whether or not to discard the fragment as occluding. This process is described in detail in Section 4. The second stage deals with the preparation for the pipe cutaway rendering. In this stage, the proxy cutaway geometry applied to the pipe lines is used to mark specific areas in the stencil buffer. Later on, these areas are needed to decide whether to draw the outer or the inner sides of the billboarded pipes. The description of this novel approach is given in Section 6. In the next stage, the actual rendering of the billboarded pipes takes place. Based on the pipe line nodes, the billboard quadrilaterals are constructed in the geometry shader. These quadrilaterals are then shaded to mimic the tubular shape of sewer pipes. Based on the stencil buffer values prepared in the previous stage, the shading mimics either the outer or to the inner side of a pipe. As a result of this, a cut-open pipe is drawn. More detail on this stage is given in Section 5. Finally, in the last stage, described in Section 7, sewer shafts are rendered based on cylinder mesh primitives. In the fragment shader, based on the viewing direction and the fragment normals, some fragments are discarded to reveal the shaft internals.

## 4. Terrain cutaways

In order to expose the subsurface sewer system, the terrain needs to be cut open (Stage 1 in Fig. 3). For this purpose we use a distance field approach which resembles the approach of Trapp et al. [39]. In their work, a distance field texture is generated from a road network in order to achieve an anti-aliased rendering of features. We use a distance field in combination with the camera distance to discard occluding terrain fragments.

A two-dimensional projection of the sewer network is drawn with black lines onto a white texture using a simple line drawing algorithm. Black is encoded as the 8-bit color value 0 and white as 255 on a grayscale texture. The texture size exactly corresponds to the original texture of the terrain. Therefore, texture coordinates in the original texture can also be used for sampling the distance texture. The texture is aligned with the 2D line representation and perfectly matches the sewer network in world coordinates, as in Fig. 4b. Additionally, a one-dimensional distance transform under the $L_1$ metrics is applied to this texture. The algorithm is applied horizontally and vertically [40]. The resulting distance field is depicted in Fig. 4a. It encodes the distance to the closest pipe in grayscale. The pipe itself has the value of 0, and beyond a certain distance to any pipe the value is 255. One may notice the artifacts due to the use of the $L_1$ distance that cause the zigzag effect (jittering) on the cutaway. Thus, a rough cutaway is created, which is considered an optional rule for computer generated cutaways for technical illustrations by Diepstraten et al. mentioned in the previous section. The rule is applicable for a simple cutout geometry to generate a higher level of abstraction [9]. The generation of the distance field texture can be considered a pre-processing step before the actual rendering.

Further on, the generated texture is forwarded to the graphics system and used as a distance field. For each fragment, the Euclidean distance to the camera $d_c$ is evaluated. Moreover, a lookup on the distance field texture is performed, by the fragment's texture coordinates $P_{uv}(u, v)$, to obtain the distance value $D_p(P_{uv})$ to the closest pipe. Obtaining $D_p(P_{uv})$ in the fragment program via a texture lookup results in a floating point value between 0 and 1. Broadly speaking, $d_c$ and $D_p(P_{uv})$ are used to decide whether the fragment is discarded or the terrain is drawn. The artificial thickness of the terrain has to be factored in as well, so that the terrain surface does not look completely flat. Before a fragment is actually discarded, it is evaluated whether it is part of the artificial thickness. Therefore, the two horizontal coordinates of the view direction vector $\hat{\mathbf{v}}_{3D}(x, y, z)$, the direction the camera is looking at, are used to make a two-dimensional peek in $\hat{\mathbf{v}}_{2D}(x, z)$ and perform another lookup on the distance field texture. In our system, $y$ is the height coordinate and is ignored. The peek will provide information if the current fragment is close to the not discarded terrain and the artificial thickness has to be drawn instead of actually discarding this fragment. To interactively control the width of the terrain opening during runtime, a steering parameter $w$ is used. Another steering parameter $s$ is used to control the size of the opening. Basically, with $w$, $s$ and $d_c$, a threshold $t$ is computed for each fragment with Formula 1 to control the dimensions of the opening (Fig. 5). With $D_p(P_{uv})$ below the threshold $t$, the routine will result in either discarding the fragment or drawing the artificial thickness. Note that Formula 1 is an empirically designed formula where the threshold $t$ is proportional to the opening width $w$ and where the size $s$ is supposed to decrease the influence of the camera distance $d_c$.

$$t = \frac{w(s^2 - d_c)}{s} \quad w \geq 0, \ s > 0 \tag{1}$$

This equation causes the smooth closing and opening of the terrain as the camera is moving across the scene. By default the

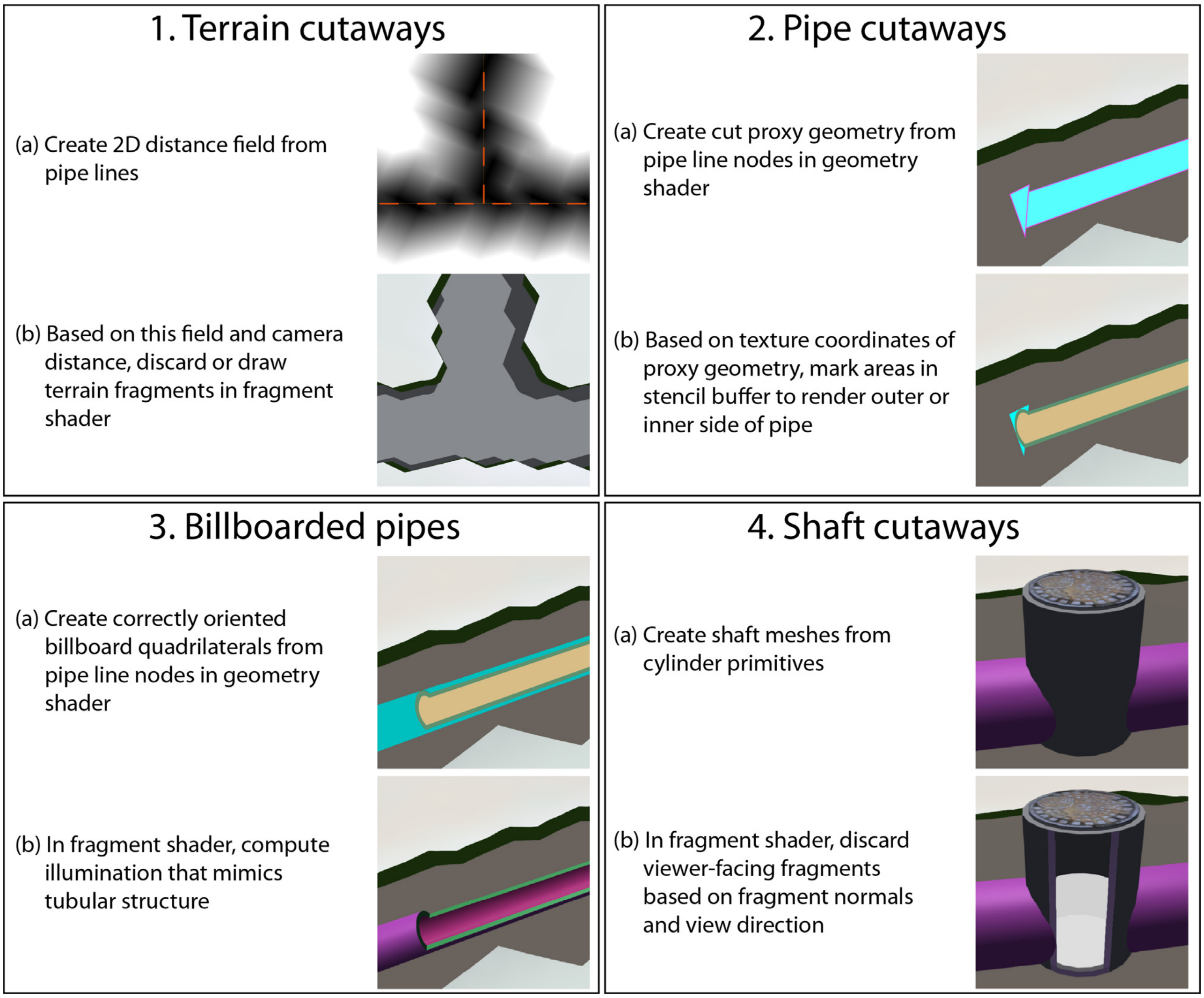


**Fig. 3.** Overview diagram of the presented approach.

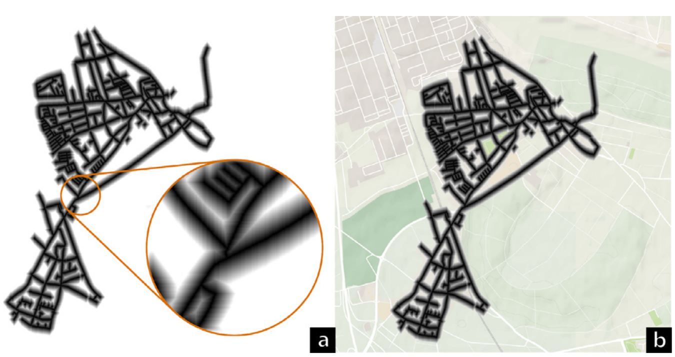


**Fig. 4.** (a) 2D distance field representation of the sewer network on white background. The artifacts at the intersections are caused by the use of the $L_1$ metrics. (b) Distance field texture perfectly aligned with the terrain texture.

parameters are set to $w = 0.015$ and $s = 10$. The following function $f_1$ is used by each fragment of the terrain to determine the final result. Here $\epsilon$ is a value big enough to obtain the next texel of the distance field texture in the viewing direction with $P_{uv} + \hat{\mathbf{v}}_{2D} \cdot \epsilon$. One suggestion would be $\epsilon = \sqrt{2}/T$ where $T$ stands for the distance

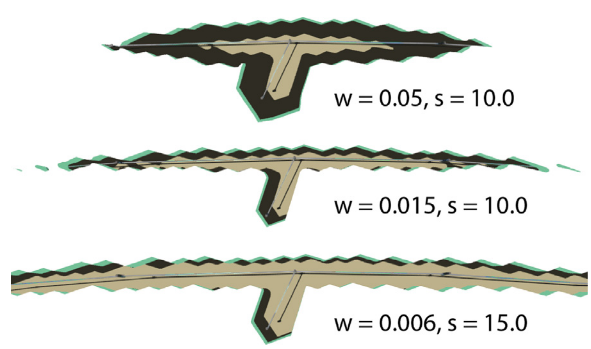


**Fig. 5.** Terrain cutaways with constant distance to camera $d_c$ and varying steering parameters $w$ and $s$ ($w$ = width, $s$ = size).

texture size.

$$f_1\left(P_{uv}, t, \hat{\mathbf{v}}_{2D}\right) = \begin{cases} \mathit{draw\ terrain} & D_p(P_{uv}) \geq t \\ f_2\left(P_{uv} + \hat{\mathbf{v}}_{2D} \cdot \epsilon, t\right) & D_p(P_{uv}) < t \end{cases} \tag{2}$$

$$f_2\left(P'_{uv}, t\right) = \begin{cases} \mathit{draw\ artificial\ thickness} & D_p(P'_{uv}) > t \\ \mathit{discard\ fragment} & \mathit{otherwise} \end{cases} \tag{3}$$

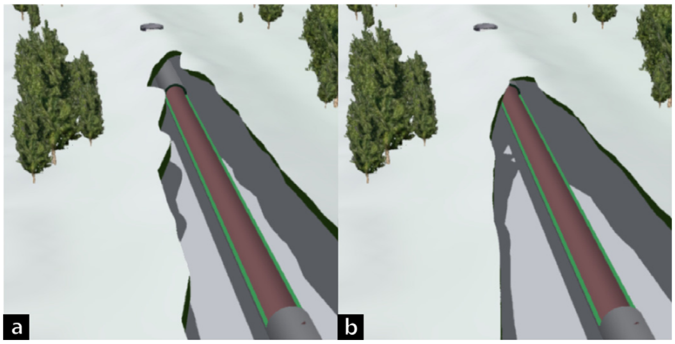


**Fig. 6.** Side-by-side comparison of a non-smooth terrain cutaway, generated using the $L_1$ metrics, with a smooth terrain cutaway, using the Euclidean metrics.

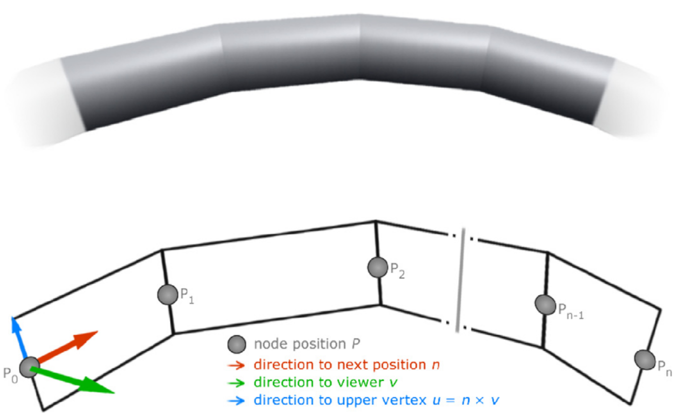


**Fig. 7.** Arrangement of quadrilaterals in a billboarded pipe. Quadrilateral vertices are constructed based on the pipe line nodes and the view direction.

Opinions differ on whether the zigzag effect should be kept or rather cleared out to get a smoother cut. The smooth cut can be achieved by implementing the squared Euclidean distance instead of the $L_1$ distance for the distance transformation. However, as figured out during the evaluation, the zigzag effect of a cutaway can sometimes be visually appealing and therefore can be kept as an optional feature. In Fig. 6, an example is shown for a smooth cutaway without the zigzag effect.

The proposed method does not support illuminated shading of the cut surface. However, this could be achieved by computing a similar texture where each pixel encodes a vector as RGB value that points in the direction of the closest pipe. This vector could be used as normal in the fragment shader and an illumination model can be calculated to shade the cut surface.

## 5. Billboarded pipe rendering

This section describes Stage 3 of the diagram in Fig. 3, where the actual billboards are created to visualize sewer pipes. In the input dataset, each sewer pipe is defined by a list of three-dimensional nodes. The first node defines the starting position in world space coordinates, and each successive node sets the spatial direction. Additionally, each pipe in the input dataset is attributed with a value for its radius. Given this data structure, we generate a visual representation of the sewer pipes using billboards. The general idea is inspired by the approach of Schirski et al. [31] and adapted to current GPU-programming practices. Instead of the standard OpenGL used in the original approach for computing the billboard vertices, we compute these in the geometry shader [41].

Components of a billboarded pipe are depicted in Fig. 7. The pipe line nodes are utilized to arrange the quadrilaterals. The node positions are wrapped in a buffer and forwarded to the geometry shader. The geometry shader calculates the required vertices of the quadrilaterals. To compute the direction to the quadrilateral's upper vertex of the first node $P_0$, the cross product of the normalized vectors $\hat{n}_0$ and $\hat{v}_0$ is used as shown in Fig. 7. The described method can be used for the last node $P_n$ of a billboarded pipe basically in the same way. For the remaining nodes, $P_1$ to $P_{n-1}$, the calculation of the corresponding direction to the quadrilaterals' upper vertices is more complex. In general, the computation of the vector $\vec{u}_i$ pointing in the direction of the upper quadrilaterals vertex for a node position $P_i$ can be described as follows:

$$\vec{u}_0 = \hat{v}_0 \times \hat{n}_0 \tag{4}$$

$$\vec{u}_i = \hat{v}_i \times \frac{\hat{n}_{i-1} + \hat{n}_i}{|\hat{n}_{i-1} + \hat{n}_i|}, \quad i \in \{1, \dots, n-1\} \tag{5}$$

$$\vec{u}_n = \hat{v}_n \times \hat{n}_{n-1} \tag{6}$$

The direction to the quadrilateral's lower vertex $\vec{l}_i$ is computed by inverting $\vec{u}_i$ for each node position $P_i$:

$$\vec{l}_i = -\vec{u}_i, \quad i \in \{0, \dots, n\} \tag{7}$$

Further on, the vectors $\hat{u}_i$ and $\hat{l}_i$ for each node position $P_i$ are multiplied by the radius $r$ of the sewer pipe, which can be changed interactively at any time. Finally, all vertex positions of the quadrilaterals are computed by translating each node position $P_i$ by the calculated vectors $\hat{u}_i$ and $\hat{l}_i$. This arrangement of billboards causes them to face the camera almost exactly for any camera position. Since one particular quadrilateral cannot use the same vectors $\hat{n}$ and $\hat{v}$ to calculate all four of its vertices, which would cause a gap between consecutive quadrilaterals, they have to be slightly twisted to line up seamlessly. This twist is achieved by calculating an average of $\hat{n}_{i-1}$ and $\hat{n}_i$ in Eq. (5). The algorithm described above corresponds to Step 3a in Fig. 3.

After the arrangement, the billboards need to be shaded. The billboards are colored, and illumination is calculated to create the impression of an actual tubular structure (Step 3b in Fig. 3). For this purpose, the normal directions at billboard vertices are required to compute the Blinn-Phong shading model, which are set by the geometry shader during their creation. However, the illumination needs to be calculated as if the flat billboards are tubular. Therefore, two vectors are set for each vertex and propagated to the fragment program. At first, the normal-direction of a billboards face is calculated and set for all four vertices of the quadrilateral. Since we do not want the slight twist to show up in the illumination, the average face normal $\vec{F}_a$ of the four vertices is determined by the following equation:

$$\vec{F}_{a_0} = \frac{\hat{u}_0 + \hat{u}_1}{|\hat{u}_0 + \hat{u}_1|} \times \hat{n}_0 \tag{8}$$

$$\vec{F}_{a_i} = \frac{\hat{u}_{i-1} + \hat{u}_i}{|\hat{u}_{i-1} + \hat{u}_i|} \times \hat{n}_i, \quad i \in \{1, \dots, n-1\} \tag{9}$$

$$\vec{F}_{a_n} = \frac{\hat{u}_n + \hat{u}_1}{|\hat{u}_0 + \hat{u}_1|} \times \hat{n}_0 \tag{10}$$

Additionally, for the upper two vertices of a quadrilateral $\vec{N}_{u_i} = \hat{N}_{a_i} \times \hat{v}_i$ is set as well as $\vec{N}_{l_i} = \hat{N}_{a_i} \times \hat{v}_i \cdot (-1)$ for the lower vertices. Here $\vec{v}$ denotes the view direction vector. The normal vectors $\hat{N}_{u_i}$ and $\hat{N}_{l_i}$ are automatically interpolated by the graphics system according to the fragments position on the billboard when retrieving this value $\hat{N}$ in the fragment shader. Since this automatic interpolation is insufficient, additionally the face normal direction $\hat{F}_a$ of a

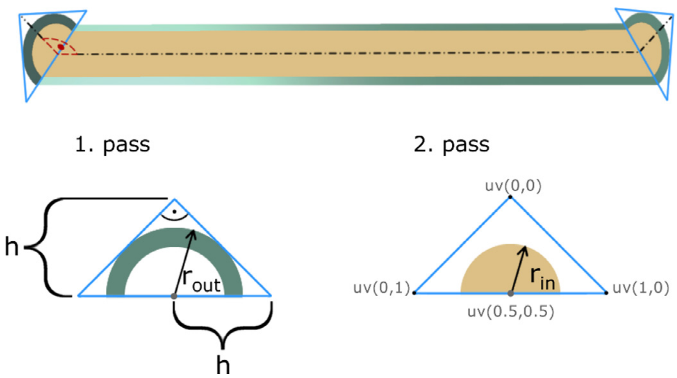


**Fig. 8.** Setting up the triangles of the proxy geometry. (top) Arrangement of the triangles vertically, perpendicular to the pipe slope. (bottom) Using the texture coordinates of the triangles to render the cut surface and to mark the areas of the stencil buffer where the inside of the pipe needs to be shown. This is done in two separate shader passes.

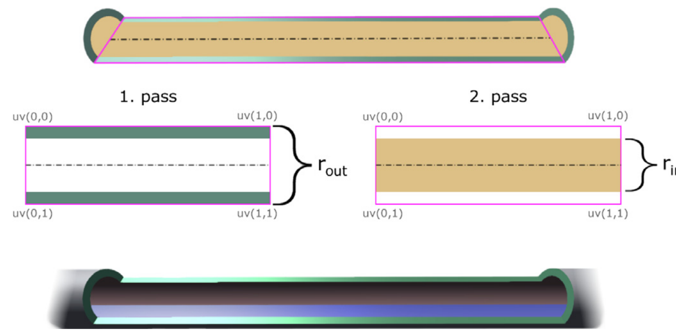


**Fig. 9.** Setting up the quadrilaterals of the proxy geometry. (top) Arrangement of the quadrilateral along the pipe. (middle) Using the texture coordinates of the quadrilateral to render the cut surface and to mark the areas of the stencil buffer where the inside of the pipe needs to be shown. This is done in two separate shader passes. (bottom) Resulting rendering of a billboarded pipe.

quadrilateral is used. To summarize, a fragment's final normal direction $\hat{N}_f$ used for illumination is determined by computing the average vector of $\vec{F_{a_n}}$ and $\hat{N}$ for each fragment as follows:

$$\hat{N}_f = \frac{\hat{N} + \hat{F}_a}{|\hat{N} + \hat{F}_a|} \tag{11}$$

The resulting normal vector mimics the illumination of a tubular structure and is further used for a standard Blinn-Phong shading model and specular highlights. In combination with the proposed non-photorealistic shading, the billboarded pipes look very similar to polygon meshes but are quite advantageous in terms of low memory footprint and faster rendering time, which is demonstrated in Section 8.

## 6. Pipe cutaways

This section contains the detailed description of Stage 2 from Fig. 3. Since the water in the pipes is the main object of interest, it has to be exposed to the viewer. In order to achieve this in combination with billboarded pipes, a novel technique is needed. We use a proxy geometry to prepare areas in the stencil buffer. These areas are used to define whether the view-aligned billboards have to be rendered and shaded for the inside or outside illumination of the pipe. Utilizing the illumination technique for the billboards described above, the billboards shading for the inside of a sewer pipe is easily done by inverting the computed normal direction in Eq. (11). Additionally, the base color is changed to enhance the visual distinction between the inner and outer parts of the sewer pipe. For the same purpose, the base color of the cut surface is chosen to contrast with any other colors in use (Fig. 8, bottom).

The general idea is to set up a static proxy geometry as demonstrated in Figs. 8 and 9 before the view-aligned billboards are applied. This will allow us to achieve two things. First, the green cut surface is rendered immediately. Second the inner area of the proxy geometry is used to set a certain value in the stencil buffer. By doing this, the area is prepared to render view-aligned billboards shaded accordingly to display the inside of the pipe. At a later stage in the rendering pipeline, the prepared area is used and the inside of the pipe is drawn.

A value written to the stencil buffer by one particular shader pass is declared for the shader pass itself. It is executed for any drawn fragment and therefore can not be changed in the fragment shader. For this reason, the described routine and further processing of the proxy geometry is carried out in two separate shader passes labeled in Figs. 8 and 9 accordingly. Important to note is that in this section, whenever a cut surface is drawn, the stencil buffer is set to a certain value. This value will prevent the outside and inside of the pipes billboard to override the cut surface.

### 6.1. Proxy geometry triangles

As demonstrated in Fig. 8, the blue triangles are created by the geometry shader based on pipe node position like the quadrilaterals of the pipe. These triangles are arranged perpendicular to the slope of the related proxy geometry segment and not changing their alignment toward the viewer. The cut surface and the area to prepare the stencil buffer are determined analytically by the use of the texture coordinates and the distance to the center of the semicircle. Therefore, the size of the triangles has to provide accurate space for the sewer pipes outer radius $r_{out}$ inside of them. This is achieved by constructing the triangle with $h = \sqrt{2} \cdot r_{out}$ and using $h$ as demonstrated in Fig. 8 when creating the vertex positions of the triangle in the geometry shader. Texture coordinates $(u, v)$ are at the same time set as depicted and used to compute the Euclidean distance $d$ to the texture coordinates (0.5,0.5) for each fragment.

$$d = \sqrt{(0.5 - u)^2 + (0.5 - v)^2} \qquad u, v \in [0, 1] \tag{12}$$

Calculating $d$ enables us to express the desired areas. The way the triangles are constructed, the outer radius $r_{out}$ can now be described by $d_{out} = 0.5$. The inner radius $r_{in}$ is expressed by $d_{in} = (\frac{r_{in}/r_{out}}{2})$. Whether the distance $d$ exceeds $d_{out}$, falls below $d_{in}$, or is in between, the fragment is handled by the following functions $T_1(d)$ and $T_2(d)$ for the first and second shader passes, respectively (in both equations, we let $d \in [0, \sqrt{0.5}]$):

$$T_1(d) = \begin{cases} draw\ cut\ surface & d_{out} > d > d_{in} \\ discard\ fragment & \text{otherwise} \end{cases} \tag{13}$$

$$T_2(d) = \begin{cases} prepare\ stencil\ buffer & d < d_{in} \\ discard\ fragment & \text{otherwise} \end{cases} \tag{14}$$

### 6.2. Proxy geometry quadrilaterals

Clearly, for the proxy geometry, more billboards are required than just the two triangles. At least one quadrilateral is needed in between. In case the cut is done on a straight part of the pipe, it comes down to only one quadrilateral. In case of a bended pipe, the quadrilaterals of the proxy geometry are computed similarly to the view-aligned billboarded pipes. The only difference is that, instead of the view direction, a vector pointing upward, perpendicular to the slope of each segment, is used.

A procedure very similar to the one described above is applied to the quadrilateral part of the proxy geometry. As shown in Fig. 9,

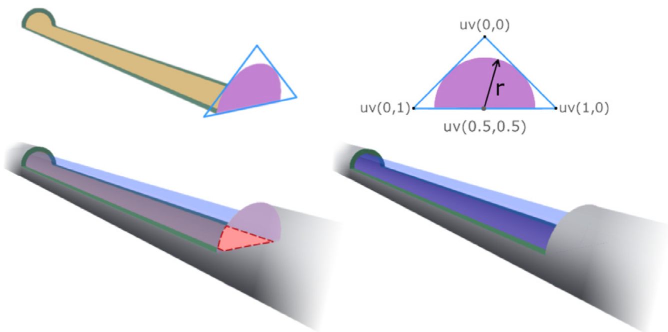


**Fig. 10.** Visual inconsistencies at the back faces of triangles (pink area marked with a red dashed line) are resolved by correctly adjusting the stencil buffer for the area marked in violet.

the pink quadrilateral is arranged to cut the pipe in half horizontally and share vertex positions with the triangles. In contrast to the method for the triangles, here the width of the quadrilaterals equals to the outer radius $r_{out}$ of the pipe. Again, the texture coordinates are used to analytically determine the two distinct areas. These texture coordinates are set as shown in Fig. 9. Since the quadrilaterals are symmetric relative to their primary axis going along the pipe, only the second value $v$ of the texture coordinates $(u, v)$ is used for these calculations. As before, the outer and inner radius, $r_{out}$ and $r_{in}$ are used as a threshold to define the desired areas on the quadrilateral. The wall thickness $t$ of the pipe is actually used and calculated by subtracting the inner radius $r_{in}$ from the outer radius $r_{out}$. Since the wall thickness value needs to be represented in the texture space, it is further divided by $r_{out}$, which results in $t = \frac{r_{out}-r_{in}}{r_{out}}$. Based on a fragments texture coordinate $v$, it is determined whether we need to draw the cut surface or set a value in the stencil buffer to declare it inside as follows by the functions $Q_1(t, v)$ and $Q_2(t, v)$ for the first and second shader passes, respectively (in both equations, let $t, v \in [0, 1]$):

$$Q_1(t,v) = \begin{cases} prepare\ stencil\ buffer & (1-t) > v > t \\ discard\ fragment & \text{otherwise} \end{cases} \tag{15}$$

$$Q_2(t,v) = \begin{cases} draw\ cut\ surface & \begin{matrix} if & v > (1-t) \\ or & v < t \end{matrix} \\ discard\ fragment & \text{otherwise} \end{cases} \tag{16}$$

To prevent visual inconsistencies, the back faces of the same triangles, as first mentioned in this section, need to exchange the stencil buffer value when facing them. The visual inconsistency that would occur is demonstrated in Fig. 10. For the violet area, a new value is set in the stencil buffer, which prevents the inside view-aligned billboards to be rendered there, and ensures to display the outside of the sewer pipe instead. Otherwise the red-dotted area in Fig. 10 would display the inside of the pipe, although it should not, given the way we are facing the cut.

### *6.3. Treatment of visual inconsistencies within the cuts*

The above mentioned step concludes setting up the image space with values in the stencil buffer in order to specify whether to render view-aligned billboards that are shaded to display inside or outside of the sewer pipe (Stage 2 in Fig. 3). Finally, the view-aligned billboards are drawn according to the prepared areas in the stencil buffer, as described in the previous section. However, the current result, as shown in Fig. 11 (top), exhibits an obvious flaw. It is apparent that the orange, red-dotted area was not prepared appropriately. The reason is that it is easier and cheaper to simply discard this area right away on the view-aligned billboards than to use the stencil buffer for this as well. With the use of texture

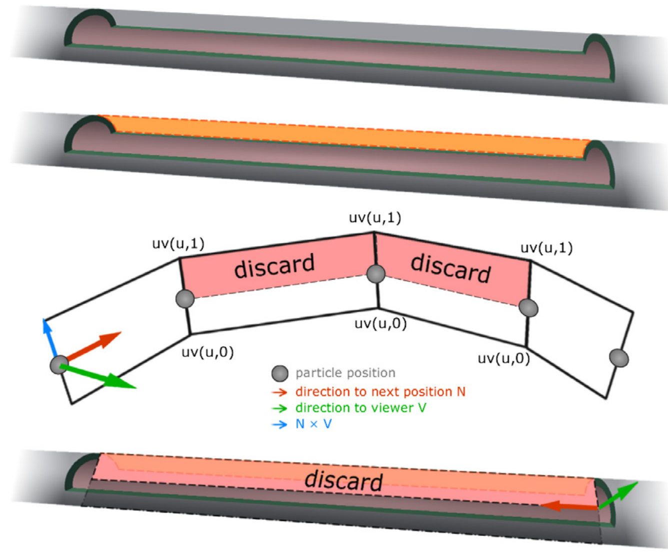


**Fig. 11.** Visual inconsistencies at cut-open pipe segments (top) are resolved by discarding unnecessary fragments in the fragment shader (bottom).

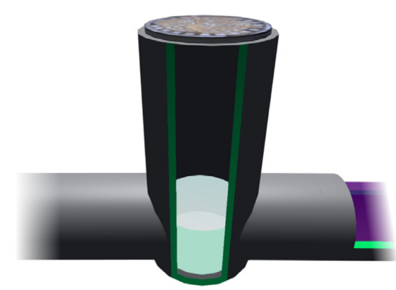

**Fig. 12.** Polygon meshed shaft constructed from cylinder primitives. The cutaway effect is achieved by discarding fragments based on the dot product of the fragment's outer normal and the view direction.

coordinates, the upper halves of billboards inside a cut-open segment are discarded in the fragment shader. Alternatively, it is possible to solve this problem in the geometry shader by not translating the upper vertices of the quadrilaterals in the cut-open pipe segment and thus not even creating this area for the view-aligned billboards.

### *6.4. Water level rendering*

The water level, given per pipe, is rendered in the last step. For each pipe opening, a procedural billboard is created and shaded appropriately to color-code the water level value. The creation procedure is similar to the one for the cutaway proxy geometry.

## 7. Shaft cutaways

This section corresponds to Stage 4 in Fig. 3. Sewer shafts are constructed from multiple cylinder primitives and are therefore made of polygon meshes. The shaft itself is constructed by an outer and an inner cylinder, where the inner one is slightly smaller than the outer one. For both cylinders, fragments are discarded when the dot product of the normal vector and view direction vector surpasses a certain coefficient. In this process, the height coordinate of both vectors is ignored. This cutaway is thereby view-dependent and the opening will always face the viewer (Fig. 12). The coefficient defines the opening width of the cutaway. Application of the cut surface color is achieved by using the same shader for back-facing triangles of the outer cylinder and the front-facing triangles of the inner cylinder. The coefficient defines the opening width of the cutaway.

## 8. Results and evaluation

In this section, we evaluate the proposed technique in terms of quality and performance. First, we evaluate our technique with respect to the five design principles for cutaways mentioned in Section 2. Further an informal review of the resulting visualizations by an architectural visualization expert and a flood management expert is given. Finally, the performance is measured in terms of the time needed to render a frame. For several classes of objects in the scene (terrain, sewer pipes, etc.) the time is measured separately. In addition to billboarded sewer pipes, the timings for meshed sewer pipes are provided.

### *8.1. Rule-based evaluation*

Rule (1) from Section 2 applies to our implementation of pipe cutaways since distinguishable colors are used for the inside and outside of the pipe. In fact, the choice of colors is up to the user of the visualization. Sewer pipes are cut open by an object-aligned box cut as the cutting convention by Li et al. [26] suggests. This satisfies Rule (2), as a box can indeed be described by a few half spaces. In accordance with Rule (3), the cutout is located at or around the main axis of the pipe. Rule (5) recommends to make the cut surface visible. In our implementation, the cut surface is visible and emphasized with a high-contrast color. For the cutaway of the sewer shafts, the same rules as for the pipes are applicable. However, a different cutting convention by Li et al. is used. The convention is described as wedge cut, since it removes the shape of the wedge from the shafts cylinder. Rule (4) suggests an optional jittering effect. In our implementation, this is not applied to the sewer pipes or shafts but is used for the terrain cutaway. The cut surface of the terrain cutaway has been made visible as well and it is arguable that it is aligned with the sewer system. Thereby Rule (3) and Rule (5) are fulfilled. Certainly, the cutout geometry used for the terrain can not be represented by a few half spaces, neither is it located around the main axis of the terrain. In fact, in our specific case where the outside object is the terrain surface, Rules (2) and (3) are hardly applicable, since the terrain is not compact and it is generally impossible to identify its axes. An alternative way would be to position the cutaway along the main axis of the *inside* object (pipe), which we do. According to the positive experts' feedback, given below, this does not hinder the effectiveness of the visualization.

### *8.2. Expert feedback*

An architectural visualization expert from *Office Le Nomade* in Vienna (Austria), the author of the inspirational image shown in Fig. 13, reviewed the screenshots and the video of our results. According to his impression, the visualization of the sewer system is performed very well. Spatial relationships of all structures are easy to comprehend. Furthermore, due to the color emphasis of the cut surfaces, it is apparent if something was cut away to reveal the interior. Therefore, it is easy for the viewer to mentally reconstruct missing parts. Regarding the zigzag effect of the terrain cutaway, the expert mentioned that on a picture or a video he would prefer the zigzag effect to be cleared out. However, after watching the video where the camera moves across the scene, the expert liked the effect of the terrain breaking up similarly to a tectonic shift. In his opinion, the biggest flaw of the implementation is the shading of the buildings and the terrain, which should be more detailed. In addition, he mentioned that he would prefer the cut surface of the terrain to have a less saturated color. In his opinion, the color

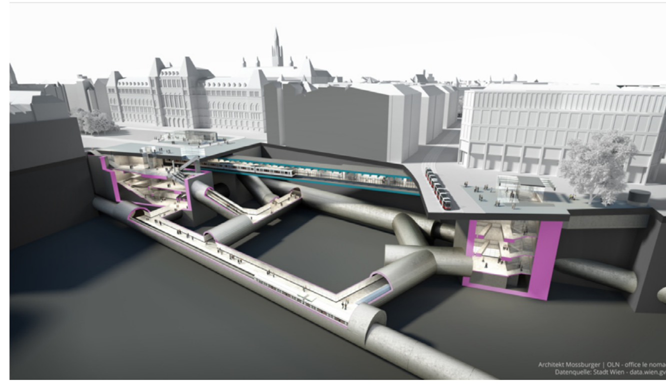


**Fig. 13.** Inspirational visualization of the underground station U5 Rathaus in Vienna (Austria), created by the architectural visualization experts from *Office Le Nomade*.

could be slightly darker than the terrain color. From his perspective, our approach for cutaways of tubular structures could be used for architectural visualizations only in very special cases.

One more expert who reviewed our results was a water resource engineer from a consulting agency agency *Riocom*, situated in Vienna (Austria). The expert works with risk assessment of flooding scenarios in urban and rural environments and deals with flood and stormwater simulations. The expert highly valued our visualization. In his opinion, this visualization gives engineers an opportunity to study the flow behavior on the surface and under the ground coherently. According to the engineer, this is crucial in modern flood management. Additionally, this visualization can successfully serve presentation purposes.

### *8.3. Performance measurements*

The example implementation was done in Unity3D, where only the standard SDK tools were used apart from the algorithm-specific shader code and the code to import and pre-process the input data. The performance has been measured on a real-world data for the city of Cologne in Germany (Figs. 14 and 15). The machine used for performance measurements was a Windows 10 x64 desktop computer with an Intel Core i7-4770 processor running at 3.4GHz stock frequency and 16GB of RAM. The graphics card was an NVIDIA GeForce GTX 1070 Ti. The whole dataset of the sewer system consisted of 1190 sewer pipes with an average of 4.6 node positions per pipe. In addition it contained 1156 sewer shafts with exactly 2 nodes per shaft. The camera was positioned in a way that the whole sewer system is rendered. Performance measurements were taken for a standalone executable in a full screen mode. Rendering time of several features was measured separately. The measured features include terrain, shafts, billboarded pipes, and meshed pipes of the same configuration for comparison. The rendering time of an empty scene is given as a baseline. For each feature, time measurements were accumulated over 10000 frames and then averaged. Measurements were taken for two screen resolutions ("L" stands for $1280 \times 960$ and "H" for $1920 \times 1200$) and two default quality levels ("0" is minimum and "5" is maximum, in Unity3D called "Fast" and "Fantastic", respectively). The quality levels in Unity3D contain presets for anti-aliasing, shadows, texture quality, reflections, and the like. The results in milliseconds are given in Table 1.

The table shows that the billboarded sewer pipes are rendered more efficiently compared to the meshed sewer pipes. Additionally, given a drastically smaller amount of polygons needed for billboarded pipes, the memory footprint is clearly lower.

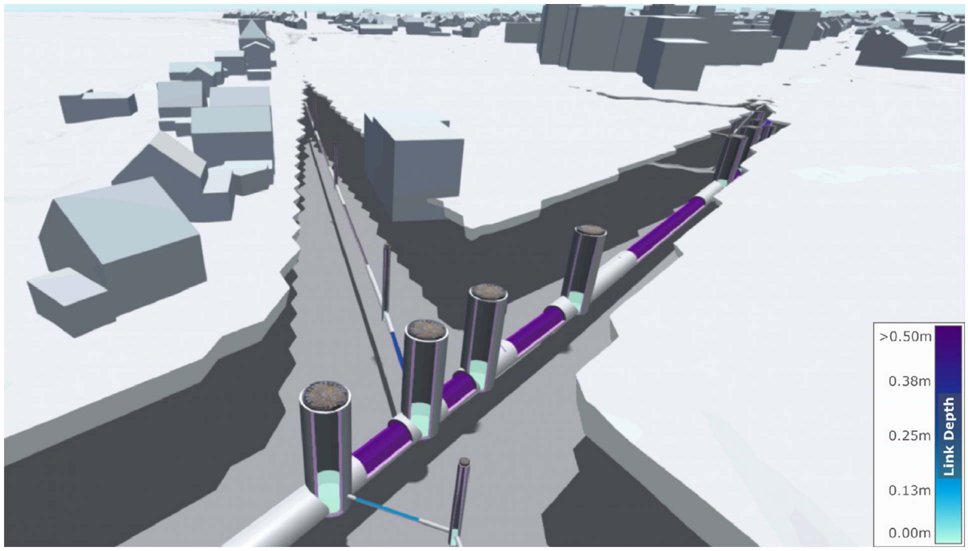


**Fig. 14.** Resulting visualization for a real-world test case considering the city of Cologne (Germany).

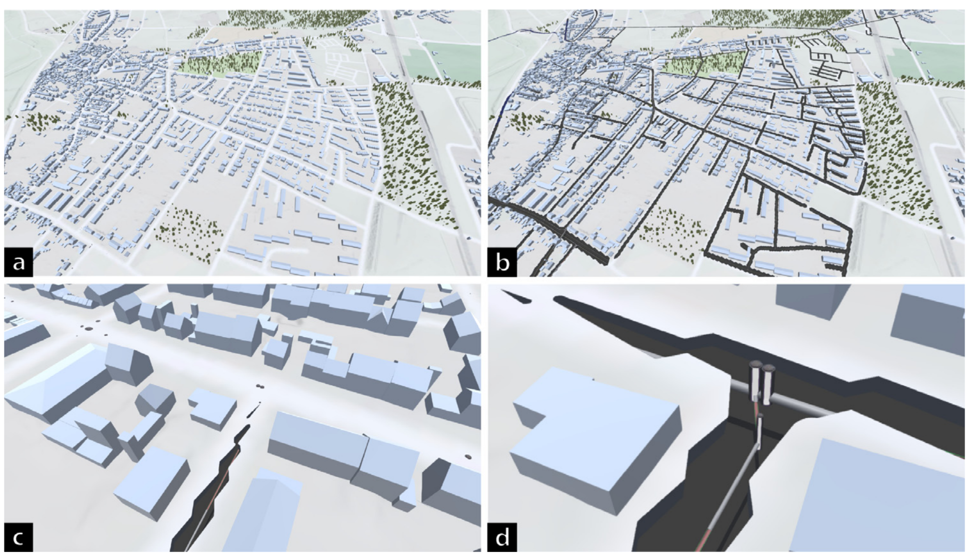


**Fig. 15.** Selected frames demonstrating the dynamic cutaways. (a) View of a district of Cologne (Germany). (b) If the camera distance dependency is disabled, it is possible to see the whole sewer network simultaneously. (c) As the user zooms in, the terrain smoothly opens to reveal the sewer network underneath. (d) In a close-up view, the terrain is cut-open, and the sewer pipes and shafts are visible. The cutaways in the pipes and shafts enable the user to see the otherwise occluded water, if any.

**Table 1**
Rendering time in milliseconds for different features. “Pipes (B)” stands for billboarded pipes, “Pipes (M)” corresponds to meshed pipes, given for comparison. “#Pol.” stands for the number of polygons, “#Vert.” for the number of vertices. “L” denotes the 1280x960 resolution, and “H” denotes the 1920x1200 resolution; “0” stands for the lowest quality level, and “5” for the highest one.

| Feature | #Pol. | #Vert. | L,0 | H,0 | L,5 | H,5 |
|---|---|---|---|---|---|---|
| Empty | 0 | 0 | 0.43 | 0.52 | 0.58 | 0.59 |
| Terrain | 2.9M | 2.2M | 3.33 | 7.23 | 3.49 | 7.55 |
| Shafts | 1.1M | 1.2M | 7.95 | 10.12 | 8.19 | 11.16 |
| Pipes (B) | 4.3K | 17.1K | 0.57 | 0.60 | 0.59 | 0.66 |
| Pipes (M) | 0.9M | 0.6M | 0.72 | 0.79 | 0.76 | 0.87 |

## 9. Conclusions and future work

In this paper, we presented a cutaway-based technique to visualize subsurface networks consisting of tubular structures in the geospatial environment. The technical contributions are a new distance-based cutaway formulation and the cutaway integration with procedural billboarding of tubular structures. The technique satisfies most of the established design principles for cutaway visualizations and is efficient enough for interactive 3D navigation at high frame rates. The view dependency of cutaways facilitates the exploration of the features of interest. Due to the analytical calculations in use, our technique allows to interactively change the radius, wall thickness and colors of the sewer pipes without expensive recalculations of meshed geometry. The presented visualization received a generally positive feedback from the experts in the fields of architectural visualization and flood management.

One downside of our technique to render billboarded sewer pipes is its inapplicability for sharply bended tubes. Such cases would cause them to look flat, and the impression of spatial capacity would be lost. Luckily, sharp bends in sewer pipes would cause poor water flow rates. For this reason, they do not usually bend in angles that could negatively impact the visual impression of the billboards. However, if this happens, a possible workaround would be to replace the sharply bended parts of a billboarded sewer pipe with a polygon meshed conjunction. Another issue can occur at the intersections of cut open pipe segments. In such cases, visual inconsistencies occur because the prepared ares in the stencil buffer are interfering with each other. To prevent this, such segments can be rendered without a cutaway. Alternatively, a routine could be implemented to automatically choose the "more important" pipe segment of the whole intersecting set and apply the cutaway only to that one. This kind of issue only occurs once in the whole test data set.

A clear direction for future work is to resolve the aforementioned issue that occurs at intersections of cut open pipe segments. A routine to resolve visual inconsistencies at sharply bended pipes would be of use, too. Future work also includes the illuminated shading to the artificial thickness of the terrain cutaways. As any other focus-and-context approach, the described techniques would benefit from an emphasized visual distinction between the "focus" (pipes) and the "context" (terrain, buildings and other surface features). We consider it a possible way to proceed. The water level indication in the cut-open pipe segments becomes nearly useless when the distance from the viewer to the pipes increases. In such cases, color-coding of water levels is a more useful feature. In fact, the whole approach could provide a higher level of detail in a visualization system where for lower levels of detail, a decal-based indication of pipes and water levels on top of the terrain surface is used.

The proposed method has the potential to be used for visualizing the interior of any subsurface tubular geometry. In the geospatial domain, it could be used for underground traffic or mining tunnels. Other areas of applicability include, for example, the medical domain, where automatic cutaway visualizations of blood vessels could be created.

## Acknowledgements

VRVis is funded by BMVIT, BMDW, Styria, SFG, and Vienna Business Agency in the scope of COMET - Competence Centers for Excellent Technologies (854174), which is managed by FFG. This work is also supported by the Vienna Science and Technology Fund (WWTF) through project VRG11-010. We thank the Stadtentwässerungsbetriebe Köln, AöR and Riocom.

## Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cag.2018.07.004.

## References

[1] Hodges ER. The guild handbook of scientific illustration. John Wiley & Sons; 2003.
[2] Jansson B. City planning and the urban underground. Oxford, Oxfordshire: Pergamon Press; 1978. Caption title.
[3] Burns M, Haidacher M, Wein W, Viola I, Groeller E. Feature emphasis and contextual cutaways for multimodal medical visualization. In: Proceedings of the EuroVis, 7; 2007. p. 275–82.
[4] Kimble D. David Kimble's cutaways: the techniques and the stories behind the art. CarTech, Incorporated; 2015. ISBN 9781613251737.
[5] John Perez Graphics & Design LLC. https://www.geoart.com/ (last visited on May, 18th 2018); 2018.
[6] London's hidden tunnels revealed in amazing cutaways. https://londonist.com/london/transport/london-cutaways (last visited on May, 18th 2018); 2018.
[7] Seligmann DD, Feiner S. Automated generation of intent-based 3d illustrations. SIGGRAPH Comput Graph 1991;25(4):123–32.
[8] Feiner SK, Seligmann DD. Cutaways and ghosting: satisfying visibility constraints in dynamic 3D illustrations. Visual Comput 1992;8(5–6):292–302.
[9] Diepstraten J, Weiskopf D, Ertl T. Interactive cutaway illustrations. In: Computer graphics forum, 22. Wiley Online Library; 2003. p. 523–32.
[10] Viola I, Kanitsar A, Gröller ME. Importance-driven volume rendering. In: Proceedings of the conference on Visualization'04. IEEE Computer Society; 2004. p. 139–46.
[11] Bruckner S, Gröller ME. VolumeShop: an interactive system for direct volume illustration. In: Proceedings of the IEEE visualization 2005; 2005. p. 671–8.
[12] Ropinski T, Steinicke F, Hinrichs K. Interactive importance-driven visualization techniques for medical volume data. In: Proceedings of the vision, modeling, and visualization (VMV); 2005. p. 273–80.
[13] Bruckner S, Grimm S, Kanitsar A, Gröller ME. Illustrative context-preserving volume rendering. In: Proceedings of the EuroVis; 2005. p. 69–76.
[14] Krüger J, Schneider J, Westermann R. ClearView: an interactive context preserving hotspot visualization technique. In: IEEE Trans Visual Comput Graphics (Proc IEEE Visual 2006), 12(5); 2006. p. 941–8.
[15] Straka M, Cervenansky M, Cruz AL, Köchl A, Šrámek M, Gröller ME, et al. The VesselGlyph: focus & context visualization in CT-angiography. In: Proceedings of the IEEE visualization; 2004. p. 385–92.
[16] Krüger A, Tietjen C, Hintze J, Preim B, Hertel I, Strauß G. Interactive visualization for neck-dissection planning. In: Proceedings of the seventh joint eurographics / IEEE VGTC conference on visualization; 2005. EUROVIS'05. p. 295–302.
[17] Kubisch C, Tietjen C, Preim B. GPU-based smart visibility techniques for tumor surgery planning. Int J Comput Assist Radiol Surg 2010;5(6):667–78.
[18] Mensmann J, Ropinski T, Hinrichs KH. Interactive cutting operations for generating anatomical illustrations from volumetric data sets. Journal of WSCG 2008;16:89–96.
[19] Birkeland A, Viola I. View-dependent peel-away visualization for volumetric data. In: Proceedings of the 25th spring conference on computer graphics; 2009. p. 121–8.
[20] Wang L, Zhao Y, Mueller K, Kaufman A. The magic volume lens: an interactive focus+context technique for volume rendering. In: Proceedings of the VIS 05. IEEE visualization; 2005. p. 367–74.
[21] Ropinski T, Steinicke F, Hinrichs KH. Visual exploration of seismic volume datasets, Journal Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG) 2006;14:73–80.
[22] Patel D, Giertsen C, Thurmond J, Gröller E. Illustrative rendering of seismic data. In: Proceedings of the VMV; 2007. p. 13–22.
[23] de Carvalho FM, Brazil EV, Marroquim RG, Sousa MC, Oliveira A. Interactive cutaways of oil reservoirs. Graphical Models 2016;84:1–14.
[24] Coffin C, Hollerer T. Interactive perspective cut-away views for general 3d scenes. In: 3D user interfaces (3DUI'06); 2006. p. 25–8.
[25] McInerney T, Crawford P. Ribbonview: interactive context-preserving cutaways of anatomical surface meshes. In: Advances in visual computing; 2010. p. 533–44.
[26] Li W, Ritter L, Agrawala M, Curless B, Salesin D. Interactive cutaway illustrations of complex 3D models. ACM Trans Graphics (TOG) 2007;26(3):31.
[27] Burns M, Finkelstein A. Adaptive cutaways for comprehensible rendering of polygonal scenes. In: ACM transactions on graphics (TOG), 27. ACM; 2008. p. 154.
[28] Le Muzic M, Mindek P, Sorger J, Autin L, Goodsell DS, Viola I. Visibility equalizer cutaway visualization of mesoscopic biological models. In: Computer graphics forum, 35. Wiley Online Library; 2016. p. 161–70.
[29] Lidal EM, Hauser H, Viola I. Design principles for cutaway visualization of geological models. In: Proceedings of the 28th spring conference on computer graphics. ACM; 2013. p. 47–54.
[30] Christiansen KR. The use of imposters in interactive 3D graphics systems. Department of Mathematics and Computing Science Rijksuniversiteit Groningen Blauwborgje. Vide Sci Tech Et Appl 1995.
[31] Schirski M, Kuhlen T, Hopp M, Adomeit P, Pischinger S, Bischof C. Efficient visualization of large amounts of particle trajectories in virtual environments using virtual tubelets. In: Proceedings of the 2004 ACM SIGGRAPH international conference on virtual reality continuum and its applications in industry. ACM; 2004. p. 141–7.
[32] Le Muzic M, Autin L, Parulek J, Viola I. cellVIEW: a tool for illustrative and multi-scale rendering of large biomolecular datasets. In: Proceedings of the VCBM; 2015. p. 61–70.
[33] 3D-GIS for infrastructure - Sivan Design. http://sivandesign.com/products/3dgis/infrastructure-3d-gis/ (last visited on March, 20th 2018); 2018.
[34] Cornel D, Konev A, Sadransky B, Horváth Z, Gröller E, Waser J. Visualization of object-centered vulnerability to possible flood hazards. Comput Graphics Forum 2015;34(3):331–40.
[35] Roberts GW, Evans A, Dodson A, Denby B, Cooper S, Hollands R, et al. The use of augmented reality, GPS and INS for subsurface data visualization. In: Proceedings of the FIG XXII international congress; 2002. p. 1–12.
[36] Zhang X, Han Y, Hao D, Lv Z. Arpps: augmented reality pipeline prospect system. In: Proceedings of the international conference on neural information processing. Springer; 2015. p. 647–56.
[37] Li W, Han Y, Liu Y, Zhu C, Ren Y, Wang Y, et al. Real-time location-based rendering of urban underground pipelines. ISPRS Int J Geo-Inf 2018;7(1):32.
[38] Bichlmeier C, Wimmer F, Heining SM, Navab N. Contextual anatomic mimesis hybrid in-situ visualization method for improving multi-sensory depth perception in medical augmented reality. In: Proceedings of the 6th IEEE and ACM international symposium on mixed and augmented reality, 2007. ISMAR 2007. IEEE; 2007. p. 129–38.
[39] Trapp M, Semmo A, Döllner J. Interactive rendering and stylization of transportation networks using distance fields. In: Proceedings of the GRAPP; 2015. p. 207–19.
[40] Felzenszwalb PF, Huttenlocher DP. Distance transforms of sampled functions. Theory Comput 2012;8(19):415–28.
[41] Microsoft. DirectX Graphics and Gaming - Direct3D - HLSL - Geometry-Shader Object, online documentation. https://msdn.microsoft.com/en-us/library/windows/desktop/bb509609.aspx (last visited on March, 20th 2018); 2018.