Interactive Multi-Perspective Views of Virtual 3D Landscape and City Models

Haik Lorenz, Matthias Trapp, Jürgen Döllner

Hasso-Plattner-Institute, University of Potsdam, Prof.-Dr.-Helmert-Strasse 2-3, 14482 Potsdam, Germany,
[haik.lorenz, matthias.trapp, doellner]@hpi.uni-potsdam.de

Markus Jobst

Vienna University of Technology, Erzherzog-Johannplatz 1, A-1040 Vienna, Austria, markus@jobstmedia.at

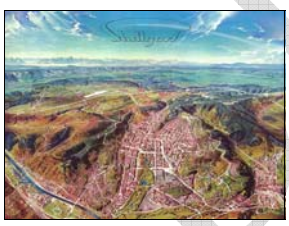

Fig. 1a. A panorama map painted by H.C. Berann (used with permission)

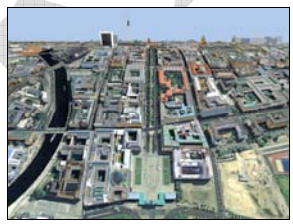

Fig. 1b. Multi-perspective view of a virtual 3D city model inspired by (a)

Fig. 1c. Multi-perspective focus & context visualization for walk-throughs

Fig. 1. A historic panorama map and examples of interactive multi-perspective views of 3D city models

Abstract

Based on principles of panorama maps we present an interactive visualization technique that generates multi-perspective views of complex spatial environments such as virtual 3D landscape and city models. Panorama maps seamlessly combine easily readable maps in the foreground with 3D views in the background – both within a single image. Such nonlinear, non-standard 3D projections enable novel focus & context views of complex virtual spatial environments. The presented technique relies on global space deformation to model multi-perspective views while using a standard linear projection for rendering which enables single-pass processing by graphics hardware. It automatically configures the deformation in a view-dependent way to maintain the multi-perspective view in an interactive environment. The technique supports different distortion schemata beyond classical panorama maps and can seamlessly combine different visualization styles of focus and context areas. We exemplify our approach in an interactive 3D tourist information system.

Keywords: multi-perspective views, focus & context visualization, global space deformation, virtual 3D city models, virtual 3D landscape models, geovisualization

1 Introduction and Motivation

Virtual spatial environments based on 3D landscape and city models are common tools for an increasing number of commercial and scientific applications and are applied as interactive space and context for planning, simulation, and visualization tasks. One key requirement represents the efficient rendering of large amounts of data based on level-of-detail techniques and multiresolution models. Another key requirement is the effective presentation of the environment and its contents, e.g., by providing detail views for important areas while giving a coarse overview of their spatial context.

While a single-perspective view depicts a scene from a single viewpoint, “a multi-perspective rendering combines what is seen from several viewpoints into a single image.” (Yu and McMillan, 2004) Mathematically, multi-perspective views rely on non-linear 3D projections or, equivalently, non-planar reference shapes, used to map 3D world space on 2D image space. In this way occlusions become resolvable, scales at which objects are depicted are adjustable, and spatial context information can be included in a single image.

With these techniques, multi-perspective views can visually emphasize or clarify an area of interest while retaining or extending its surrounding area, achieving an effective information transfer (Keahey, 1998). Furthermore, they utilize the available screen real estate to a high degree. Their characteristics make multi-perspective views a tool for focus & context visualization. Well-known examples include fisheye maps, which emphasize important information by magnification, or spherical maps, which add context information by non-uniformly integrating a full 360° view.

1.1 Multi-Perspective Views for Maps

Multi-perspective views have been developed particularly in landscape depiction and Cartography. Chinese landscape painters used multi-perspective views in the 11th century already (Vallance and Calder, 2001).

Fig. 2. Painting of Venice, Italy (about 1550) (Whitfield, 2005). It exhibits a panoramic effect and includes labels

Another example, a 360° panorama view of the London skyline consisting of six separate paintings, was created in the late 18th century. The incorporation of cartographic information yields panorama maps. Fig. 2 shows an early example of Venice, Italy (about 1550). H.C. Berann, an Austrian artist and panorama maker, pioneered one particular kind of panorama map. Beginning in the early 1930's he created a deformation and painting style (Fig. 1(a)), known as "Berann panorama", which became the de-facto standard for tourist maps in recreational areas. This style seamlessly combines a highly detailed image of the area of interest with a depiction of the horizon including major landmarks. The area of interest is shown in the foreground from a high viewpoint, whereas the horizon is shown in the

background from a low perspective. The environment is depicted with "natural realism" (Patterson, 2000) and key information such as trails or slopes is superimposed in an abstracted, illustrated fashion. As a result of the high viewpoint the foreground shows key information top-down, i.e., free from obstructions and clearly visible. At the same time, the map user can easily orient the map using the horizon, which is visible due to the changed perspective, as reference without the need for a compass. For these reasons, panorama maps are useful specifically to unskilled map readers.

In general, the creation of panorama maps is time consuming and requires a skilled artist. It includes proper viewpoint selection, partial landscape generalization, identification of landmarks, their integration into the map with recognizable shapes, and a smooth transition between the foreground and background perspective (Patterson, 2000). Even with the support of digital tools and digital 3D geodata, panorama creation still remains a tedious manual process (Premoze, 2002). Despite their effectiveness, panorama maps are rarely created, and the creation techniques can hardly be transferred to interactive systems where the user manipulates the viewpoint.

1.2 Multi-Perspective Views for Spatial 3D Environments

Multi-perspective views can be used to visualize 3D landscape models, e.g., mountainous regions with the mountain peaks providing a distinctly recognizable background for orientation purposes. Similarly, they can visualize 3D city models, using the skyline of the city as background. In today's applications, interactive visualization is required to support the user in exploring and analyzing the virtual 3D environment. With respect to the usability of such applications, the navigation and orientation aids represent key issues because users frequently "get lost in space" without guidance (Buchholz et al., 2005). Here, the inclusion of a fixed horizon or skyline similar to a Berann panorama offers an additional orientation cue in the sense of a focus & context visualization.

To obtain an automatic, real-time enabled solution, we need to focus on the projection as major tool for orientation and neglect artistic aspects such as landmark depiction and selective generalization.

Computer graphics knows three approaches to achieve a panorama effect: multi-perspective images, deformations, and reflections on non-planar surfaces (Vallance and Calder, 2001). Multi-perspective images either use non-linear, non-uniform projections or combine multiple images from different viewpoints to create the final rendering. Deformations distort the landscape before rendering the final image using a standard projection,

which implies recomputation of all geometric data for every image. Finally, reflections on non-planar surfaces use standard projections showing an intermediate object that in turn reflects the landscape.

1.3 Interactive Multi-Perspective Views

Techniques implementing multi-perspective views can be classified as multi-pass or single-pass. Multi-pass techniques create several intermediate images that are blended in a final compositing step. Each intermediate image requires separate data processing, which is rather expensive when it comes to complex spatial 3D environments. Specifically, out-of-core algorithms can incur additional penalties because rendering of intermediate images often significantly reduces caching efficiency. Additionally, image quality suffers due to resampling in the compositing step. Single-pass techniques do not exhibit these disadvantages, yet they require customization of the rendering process available only in software rendering (e.g., ray tracing) until recently.

With the advent of a programmable rendering pipeline on GPUs the implementation of interactive single-pass multi-perspective view techniques becomes feasible. We demonstrate a technique that implements a dynamic global deformation and shifts this task to the GPU. This approach exploits best the optimization of current graphics hardware for standard projections both in terms of image quality and speed. We apply our technique to an interactive application that visualizes complex virtual 3D city models in the context of a tourist information system. Our global deformation is not only used to mimic Berann panoramas but also for a novel viewing technique that enables looking ahead the current route in a pedestrian's view.

An important aspect of this contribution is the analysis of view parameters. In contrast to an artist choosing viewpoints for map creation, users of interactive applications are inherently free to move. We analyze how to define the multi-perspective view and how to dynamically adjust our deformation accordingly during the user's navigation. In addition, we discuss the implications for common 3D navigation techniques.

The paper is structured as follows. Section 2 discusses related work. Section 3 explains techniques for interactive multi-perspective views and their use for focus & context visualization. Section 4 describes the implementation. Section 5 discusses the test application and its performance. Section 6 concludes the paper and outlines future work.

2 Related Work

The work of H.C. Berann includes maps, panoramas and fine art (Berann, 2007). His way of creating panorama maps and techniques are described in (Patterson, 2000). (Premoze, 2002) introduces a first approach for implementing these techniques except for multi-perspective views by means of 3D computer graphics. Additionally, instructions for manual creation of panorama maps using various tools are available online.

Besides the artistic and visual quality of a Berann panorama, panoramic depictions use a concept known as focus & context in the field of visualization. In general, such visualization not only contains the actual subject but also its embedding context with the goal of supporting the user's interpretation process. Traditionally, focus & context has been regarded as distortion-based view of 2D or 3D information where emphasis is achieved through varying magnification and screen real estate allocation. See (Leung and Apperley, 1994) for a survey of different approaches and (Carpendale and Montagnese, 2001) for a general definition. (Vallance and Calder, 2001) presents ideas on the use of multi-perspective views for focus & context and their different creation techniques. Recently, this concept has been extended to include other methods for emphasis, such as generalization, rendering style, blur, or transparency (Hauser, 2003). (Keahey, 1998) generalizes focus & context to providing separate information dimensions.

Multi-perspective views have been analyzed mainly in the context of ray tracing, which allows for easy manipulation of the camera model. (Yu and McMillan, 2004) defines general linear cameras as affine combinations of 3 sample rays. (Löffelmann and Gröller, 1996) proposes a camera model based on arbitrary surfaces to define viewing rays and a projection surface. For real-time environments, (Yang et al., 2005) describe 3D view deformations as postprocessing step to achieve multi-perspective views and nonlinear perspective projections. (Spindler et al., 2006) improves on this method by integrating the view deformation directly into the image formation process through a camera texture. (Glassner, 2004, parts 1 and 2) describe an interesting non-interactive semiautomatic method to transfer the artistic Cubism style to computer generated images. Applications of multi-perspective views include, among others, story-telling, image processing with the goal of creating panoramas from multiple images or video footage (Roman et al., 2004), recovery of 3D information (Li et al., 2004), and image-based rendering (Levoy and Hanrahan, 1996).

Deformation is a well-established field in geometric modeling. (Barr, 1984) is one of the first describing deformation operators. Such operators are used frequently in current modeling tools. Research topics include vol-

ume preservation, avoidance of self-intersection, or deformation control. Implementations can be classified as shape deformation or space deformation. Recent examples for the former approach are (Angelidis et al., 2004; von Funck et al., 2006), which result in interactive deformations for moderately sized models. The latter approach is useful for ray casting or ray tracing. (Kurzion and Yagel, 1997) presents space deformations for hardware-assisted volume rendering.

A prerequisite for our implementation is rendering of spatial 3D environments, which includes terrain rendering (e.g., (Asirvatham and Hoppe, 2005; Hwa et al., 2004; Lindstrom and Pascucci, 2002)) and rendering of large scenes. Approaches for the latter include out-of-core algorithms (e.g., (Buchholz and Döllner, 2005; Gobbetti and Marton, 2005)), specialized visibility detection algorithms (e.g., (Chhugani, 2005; Wonka et al., 2001)), and level-of-detail algorithms (e.g., (Sander and Mitchell, 2006)). Additionally, interaction and navigation within a spatial 3D environment is necessary. (Buchholz et al., 2005) contains both, a survey of navigation techniques and improvements to common navigations.

3 Effective Presentation of Spatial 3D Environments

Multi-perspective views facilitate the implementation of effective presentation of spatial 3D environments. They can add valuable cues by seamlessly integrating multiple perspectives in the resulting images and, therefore, make efficient use of the image space.

In the following, we present two related deformation techniques that implement multi-perspective views:

1. The bird's eye view deformation, which mimics Berann's panorama maps used to visualize mountain areas.
2. The pedestrian's view deformation, which swaps the role of foreground and background by presenting a low altitude perspective view in front of a top view of distant city parts.

In general, both deformations need to ensure the user's location awareness during navigation and interaction. Even experienced users get disoriented if the current perspective does not contain sufficient points of reference or if the image sequence does not provide spatio-temporal coherence. For these reasons, both techniques provide a seamless combination of different views in a single image and achieve interactive frame rates.

We describe both deformation techniques using a reference plane *T*, a usually horizontal plane. This plane can be elevated, e.g., to define the roof

of the average building as the horizon or to reduce distortion artifacts. A point P of the virtual 3D city model not lying in that reference plane is assigned a reference point P_T in T. Deformation is then calculated using P_T and applied to P. P_T can be either a simple vertical projection of P onto T or – if an object's shape is to be kept free from distortion – a single reference point for the whole object.

3.1 Bird's Eye View Deformation

Fig. 3. The bird's eye view deformation shows a top view and the horizon simultaneously

Similar to Berann's panorama maps, this deformation is based on

- a depiction of the area of interest using a bird's eye view, which would not permit a visible sky,
- a view of the horizon and sky, and
- a smooth transition between both perspectives.

As a result, the landscape appears to be separated into two planar sections connected by a curved transition zone with the focus lying on the bird's eye view part in the foreground or lower image part (Fig. 3). Nevertheless, the area of interest is not strictly separated from the transition zone but often reaches into the curved section.

For a painted panorama map, the map designer decides on relevant parameters such as the two view points and the transition in between. In an interactive application the user can move the camera. To keep the three key properties of this multi-perspective view regardless of the camera's

orientation or position, we define fixed image areas separated by horizontal lines for the bird's eye view, transition zone, and horizon (Fig. 4). This fixation results in a transition zone curvature that depends on the viewing angle, yet the fixed horizon provides strong temporal coherence and eases orientation tracking during navigation, whereas the ever-changing shape of the landscape does not lead to distraction. In addition, this implicit definition of the horizon's perspective permits the user to navigate relative to and interact with the focus area using standard metaphors for virtual environments while the visual context is adjusted automatically.

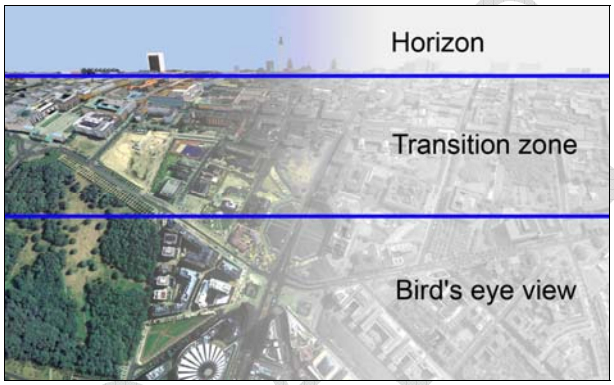

Fig. 4. Fixed image separation for the bird's eye view deformation

In general, painted panoramas exhibit a horizontal horizon. In contrast, an interactive application can permit rolling of the camera. In this case, the horizon should provide feedback about the roll angle. In the following descriptions, we assume no rolling.

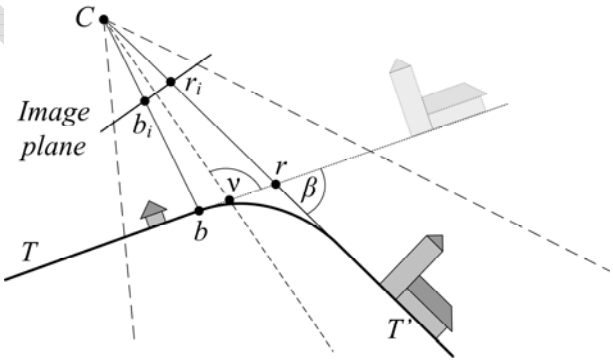

Fig. 5. Schematic side view of the bird's eye view deformation

The image subdivision results in the following set of viewing parameters:

- C – camera position
- v – viewing angle of the reference plane
- b_i – line separating focus area and transition zone in the image
- r_i – line of the horizon in the image

Fig. 5 sketches a typical setting assuming a perspective projection. In our implementation the transition zone is guided by a quadratic Bézier spline due to its continuity properties at the borders. The exact computation is described in Section 4. The line b – the projection of b_i onto T – marks the beginning of the transition zone. The half-plane following the transition zone, which leads to the horizon depiction, is referred to by T'. It is computed as a rotation of T by an angle β about the line r, the projection of r_i onto T. Thus, an object's shape is maintained outside the transition zone. Within this zone an object's shape is preserved only if it uses a single reference point.

Typically, both b_i and r_i are rarely changed while C and v reflect the user's navigation. To make efficient use of the screen space, the amount of visible sky should be minimized while retaining a recognizable skyline. Placing r_i in the upper quarter of the screen generally gives good results. The location of b_i determines the curvature of the transition zone. Placing b_i in the lower half of the screen gives a good compromise between smooth transition and visibility of the focus area.

3.2 Pedestrian's View Deformation

Fig. 6. The pedestrian's view deformation combines a realistic view of the user's vicinity with a top view of distant areas

The bird's eye view deformation supports answering questions such as "Which direction am I looking to?" without the need for a compass. For pedestrian's views, which occur in walk-through scenarios, the question changes to "Where am I going to?", e.g., if users want to look ahead the path along they are currently walking. Due to the low viewing angle, however, users can generally not obtain an effective overview without changing the perspective or navigation mode because large parts of the spatial 3D environment are occluded.

To counter this effect, the pedestrian's view deformation bends upwards distant parts of the reference plane (Fig. 6). Compared to the technique proposed in (Vallance and Calder, 2001), which deforms the reference plane to fit the inside of a cylinder, the pedestrian's view deformation has the advantage of using a planar and, hence, clear and undistorted view of distant regions in the background. In terms of focus & context, the prominent sky in a pedestrian's view, which provides only little information, is replaced by a top view of the region ahead, resulting in a more efficient use of screen space.

Similar to the bird's eye view deformation, the landscape is separated into two planar sections connected by a curved transition zone, yet the image-based deformation definition is not appropriate as it does not lead to comprehensible context behavior. We observe a more effective visualization with the Pedestrian's view deformation when using a fixed orientation of T' relative to the reference plane T in world space. Particularly, this enables an intuitive "looking-up" operation to reveal more of the context information in the background. As a consequence, the curvature of the transition zone does not depend on the viewing angle but can be defined independently. Nevertheless, the deformation follows the camera such that the rotation axis r has a fixed distance and orientation relative to the camera.

With this definition, again the user is relieved from explicitly controlling the multi-perspective view. Standard navigation metaphors within the foreground remain applicable. Only interaction with the background, e.g., for the click-and-fly navigation (Mackinlay et al., 1990), needs to be aware of our deformation for correct object identification.

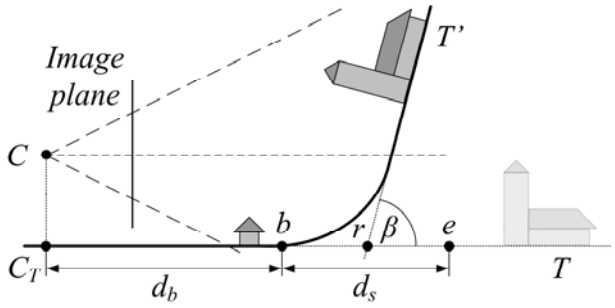

Fig. 7. Schematic side view of the pedestrian's view deformation

We use the following set of parameters to specify this multi-perspective view (cp. Fig. 7):

- C – camera position
- β – angle between T and T'
- d_b – distance between C_T (C projected onto T) and b
- d_s – width of the transition zone's source area

Analog to the bird's eye view deformation T' is a rotation of T about r and the transition zone follows a quadratic Bézier spline. The line b marking the beginning of the transition zone is always parallel to the image plane and keeps a distance d_b from the camera's vertical projection C_T onto T. We define the line r to be the center line of the transition zone's source area. Thus, it is parallel to b at a distance of d_s / 2. This definition simplifies the implementation shown in Section 4.

The parameters except for C again change rarely. They control two main characteristics of the pedestrian's view deformation: the amount of available orientation reference in the focus area through d_b and the amount of visible context information through β. Setting β to values less than 90° trades magnification in the context area for visible space and allows for looking ahead a route farther. The parameter d_s directly controls the transition zone's curvature where a small transition zone and thus rather high curvature shows good results.

3.3 Graphical Representation of Focus and Context

Both deformations presented in Section 3 smoothly and seamlessly combine focus and context. Due to the view dependent nature of both deformations, the user might loose distinction between geometrically correct information in the focus area and deformed information in the context during

navigation. This might lead to misinterpretations, lost orientation, or erroneous navigation (Zanella et al., 2002). Specifically, the pedestrian's view deformation, permitting views without visible focus area and, hence, without navigation reference, is prone to such effects. Solutions require visual cues, e.g., iconic navigation aids or distinct rendering styles for focus and context such as context color desaturation.

Besides the more effective use of screen space, in focus & context visualization the two constituents can serve different purposes and thus are to display different information dimensions beyond change of rendering style (Keahey, 1998; Stone et al., 1994). Whereas the focus gives core information, the context shows supporting information.

Panorama maps as inspiration for our bird's eye view deformation use this principle by adding thematic information such as trails to the focus area while the landscape depiction style is constant for the whole image. We demonstrate an extension showing a map with 3D landmarks in the focus area. The context remains a complete and photorealistic depiction since the skyline is required to be recognizable. Nevertheless, generalization techniques such as (Döllner et al., 2005) might prove useful. Additionally, context information can be enriched by labeling landmarks as seen in some of Berann's panorama maps.

The pedestrian's view deformation permits displaying more important information in the context. In fact, the focus area is limited to serve as navigation reference and location marker within the spatial 3D environment whereas the context generally receives the larger screen space and exhibits less occlusion. According to this observation, our sample visualization (cp. Fig. 1(c)) shows a photorealistic view in the focus and a map as context for visual distinction. On top, the current travel route is highlighted spanning both parts and thus allowing for a route preview.

Rendering such composite depictions does not require multi-pass techniques. Instead, the deformation implementation presented in Section 4 provides a vertex-based interpolation value $q \in [0;1]$ with $q = 0$ within the focus area, $q = 1$ within the context area, and a smooth transition in between. The rendering styles are then interpolated per pixel based on this value q.

4 Real-Time Deformation Implementation

The implementation shifts the deformation task to the GPU. Changing geometry on the GPU, however, has major consequences for standard application-based rendering optimizations such as occlusion culling or view frustum culling.

Our deformation scheme does not introduce new vertices. Rendering artifacts due to insufficient tessellation can only appear in the curved transition zone. This confined nature allows for a straightforward solution: a Level-of-Detail algorithm selects a more detailed object representation within the transition zone. Alternatively, on-demand tessellation using techniques such as generic mesh refinement (Boubekeur and Schlick, 2005) or the newly introduced geometry shaders can be used.

The GPU is a highly parallel streaming processor, thus each vertex needs to be processed independently. This is achieved by formulating the deformation of an individual point P as a function $f_D : P \mapsto P'$ which is computed by a vertex program. For efficient computation we want this function to perform an affine transformation M_D on P, where the 4x4-transformation matrix M_D depends only on the reference point P_T. Thus, we reformulate f_D as $P' = M_D(P_T) \cdot P$. Both deformations described in Section 3 share the same underlying construction, allowing us to use a single function $M_D(P_T)$.

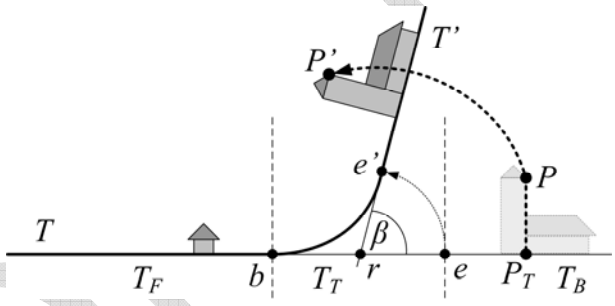

Fig. 8. Deformation parameters and definitions. Only objects located in T_T become distorted

For our unified deformation, we divide the reference plane T into three sections:

1. The undeformed part T_F , which becomes the foreground or focus of the image,
2. The transition zone T_T , which becomes curved, and
3. The remainder T_B , which becomes the background or context of the image by rotating T about r.

These three sections are separated by the line b between T_F and T_T and the line e between T_T and T_B. The lines b, r, and e are parallel and equidistant. Finally, β denotes the angle between T and T'. Fig. 8 sketches this setting.

With these three sections, $M_D(P_T)$ can be formulated depending on the location of P_T. In addition, the rendering style interpolation value q can be derived:

- $P_T \in T_F$: $M_D(P_T)$ is the identity matrix, since this section is not to be deformed; $q = 0$.
- $P_T \in T_T$: $M_D(P_T)$ needs to specify a transformation that transforms P_T and its frame of reference to the corresponding point P_T' on a quadratic Bézier spline in the tangential frame of reference. A suitable transformation consisting of a scaling followed by a rotation based on the de Casteljau algorithm (Gallier, 1999) is described in the following paragraph; q equals the Bézier spline parameter t.
- $P_T \in T_B$: $M_D(P_T)$ is a rotation matrix about r with an angle β; $q = 1$.

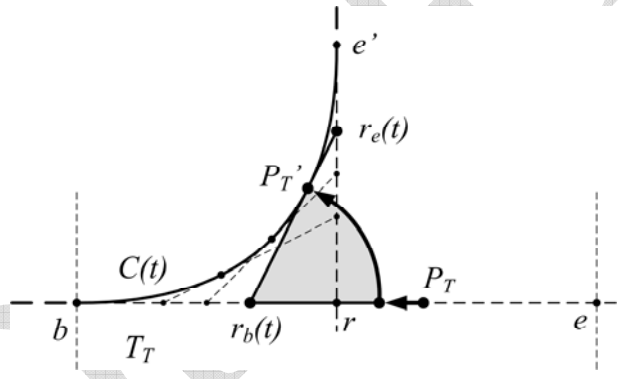

Fig. 9. The de Casteljau algorithm constructs a Bézier spline point through linear interpolations. It also provides the point's tangent

Fig. 9 shows the de Casteljau algorithm for the profile Bézier spline. The axes *b*, *r*, *e*, and *e'* appear as points in this side view, where *b*, *r*, and *e'* become control points of the spline. Since $\|b-r\| = \|e'-r\|$, the resulting spline is symmetrical. For a quadratic Bézier spline $C(t)$ with $t \in [0;1]$, the algorithm uses linear interpolations to construct two intermediate points $r_b(t) = (1-t)b + tr$ and $r_e(t) = (1-t)r + te$ and the resulting point $C(t) = (1-t)r_b(t) + tr_e(t)$.

For a given point $P_T \in T_T$, the corresponding point on the Bézier spline is found as $P_T' = C(\|P_T - b\| / \|e - b\|)$. This mapping does not define an arc length parameterization of C and thus introduces an unwanted flattening of

objects within the transition zone. The suitable reparameterization of C can be achieved using a lookup table, which is left for future work.

For our purpose, the key property of the de Casteljau algorithm is the implicit tangent construction formed by the line through r_b and r_e. To compensate for the variable length contraction along the curve, i.e., the missing arc length parameterization, a scaling along $e-b$ with a factor $\|P'_T - r_b\| / \|P_T - r_b\|$ centered at r_b is necessary. Then, a rotation of the scaled P_T about r_b onto the tangent creates the correct tangential frame of reference for P'_T. This completes the definition of $M_D(P_T)$.

The parameters T, b, e, and β of this unified computation depend on the camera location. Thus, within an interactive application, they need to be derived from the original (camera-independent) deformation parameters described in Section 3 on a frame-by-frame basis. Also, especially for scene graph based systems, the current frame of reference needs to be taken into account. The most efficient and robust solution is to perform the deformation in the camera's frame of reference since it is constant during image generation.

5 Performance

Fig. 10a. Bird's eye view deformation showing public transport lines

Fig. 10b. Pedestrian's view deformation with highlighted route

Fig. 10. Sample multi-perspective images. The insets show the corresponding standard perspectives

Fig. 10 shows sample images of the bird's eye view deformation and pedestrian's view deformation, respectively. For comparison, the insets show a standard perspective projection using identical camera settings to highlight the effects of our focus & context visualization.

We extended an existing 3D tourist information system for Berlin, Germany, with our technique. Despite the additional data handling overhead for two rendering styles, we were able to achieve interactive frame rates. Table 1 summarizes average frame rates for two sample camera paths per view deformation. The measurements were made on a PC with an AMD Athlon 64 X2 (2.3 GHz), 2 GB main memory, and a NVidia GeForce 7900GT with 256 MB video memory. The test application does not utilize the second CPU core. The sample dataset comprises the inner city of Berlin with about 16,000 generically textured buildings, about 100 landmarks, a 3 GB color aerial photo, and a 250 MB grayscale map image on top of a digital terrain model.

Table 1. Performance measurements for different screen resolutions and configurations

Resolution	Configuration	Path	frames/ sec	frames/sec without bend.
1600x 1200	Pedestrian's view	1	11.72	12.95
		2	19.33	15.69
	Bird's eye view	3	8.35	29.86
		4	6.73	17.64
1024x 768	Pedestrian's view	1	17.85	15.63
		2	22.75	17.69
	Bird's eye view	3	8.87	27.24
		4	5.42	16.07
800x 600	Pedestrian's view	1	20.54	16.49
		2	23.94	18.42
	Bird's eye view	3	8.74	27.26
		4	8.48	19.52

The frame rate without deformation is largely resolution independent suggesting texture access as main bottleneck in our test application. To deal with the texture amount, an out-of-core algorithm is used to load texture on demand in sufficient resolution from disk. Table 2 shows the average number of bytes read from hard disk per frame for our test setting at resolution 1600x1200.

Table 2. Average hard disk access per frame with / without deformation at resolution 1600x1200

Configuration	Path	bytes/ frame	bytes/frame without bend.
Pedestrian's view	1	260,207	407,822
	2	122,729	215,398
Bird's eye view	3	5,720,803	190,824
	4	2,555,602	243,067

The exceptionally high load rates for the bird's eye view deformation are caused by the visible horizon. Compared to the corresponding standard perspective projection, more terrain is visible and thus more texture requires loading – even though at low quality. At the same time, changing the view direction invalidates more texture. Hence, caching efficiency is reduced dramatically. With the pedestrian's view deformation, only a fraction of the terrain is visible compared to a standard view, but due to the deformation distant terrain requires a significantly higher texture resolution leading to only a slight reduction in texture load overhead.

6 Conclusions

We have demonstrated the concept and implementation of interactive multi-perspective views for spatial 3D environments. They are inspired by the well-known panorama maps and aim to increase the effectiveness of interactive applications by using the principle of focus & context visualization. Our implementation is based on a global space deformation processed by graphics hardware and permits the seamless combination of different graphical representations for focus and context areas. To verify its applicability we have successfully integrated our technique into an existing interactive 3D tourist information system.

The visual quality in the transition zone can be further improved by incorporation of on-demand geometry tessellation, e.g., through the use of geometry shaders, or by adaptation of a more advanced bending scheme. In contrast to the currently used simple static lighting, dynamic lighting and shadowing within a deformed 3D landscape model remains an interesting open question. While this contribution describes the underlying technology, user studies about the effectiveness and/or expressiveness of our visualization approach, different rendering style combinations, and navigation in a deformed 3D landscape model remain future work.

Acknowledgements

We would like to thank 3D Geo GmbH (www.3dgeo.de) for providing the implementation platform LandXplorer, an authoring and presentation system for virtual 3D city models and landscape models. We also thank Matthias Troyer for his permission to use the panorama map shown in Fig. 1(a).

References

The world of H.C. Berann (accessed 2007), url: http://www.berann.com

Angelidis, A., Cani, M.-P., Wyvill, G. & King S. (2004), Swirling sweepers: Constant-volume modeling, *in* Proceedings of the 12th Pacific Conference on Computer Graphics and Applications, IEEE Computer Society, Washington, DC, USA, pp. 10-15.

Asirvatham, A. & Hoppe, H. (2005), Terrain Rendering Using GPU-Based Geometry Clipmaps, *in* M. Pharr (ed.), GPU Gems 2, Addison-Wesley, pp. 27-45.

Barr, A. H. (1984), Global and Local Deformations of Solid Primitives, *in* SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques, ACM, New York, NY, USA, pp. 21-30.

Boubekeur, T. & Schlick, C. (2005), Generic Mesh Refinement on GPU, *in* Proceedings of ACM SIGGRAPH/Eurographics Graphics Hardware 2005, ACM, pp. 99-104.

Buchholz, H.; Bohnet, J. & Döllner, J. (2005), Smart and Physically-Based Navigation in 3D Geovirtual Environments, *in* IV '05: Proceedings of the Ninth International Conference on Information Visualisation, IEEE Computer Society, Washington, DC, USA, pp. 629-635.

Buchholz, H. & Döllner, J. (2005), View-Dependent Rendering of Multiresolution Texture-Atlases, *in* Proceedings Information Visualization 2005, pp. 215-222.

Carpendale, M. S. T. & Montagnese, C. (2001), A Framework For Unifying Presentation Space, *in* UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology, ACM, New York, NY, USA, pp. 61-70.

Chhugani, J.; Purnomo, B.; Krishnan, S.; Cohen, J.; Venkatasubramanian, S. & Johnson, D. S. (2005), vLOD: High-Fidelity Walkthrough of Large Virtual Environments, *IEEE Transactions on Visualization and Computer Graphics* **11**(1), pp. 35-47.

Döllner, J.; Buchholz, H.; Nienhaus, M. & Kirsch, F. (2005), Illustrative Visualization of 3D City Models, *in* Robert F. Erbacher; Jonathan C. Roberts; Matti . T. Gröhn & Katy Börner, ed., Visualization and Data Analysis 2005, pp. 42-51.

von Funck, W., Theisel, H. & Seidel, H.-P. (2006), Vector field based shape deformations, *in* Proceedings ACM SIGGRAPH 2006, ACM, New York, NY, USA, pp. 1118-1125.

Gallier, J. (1999), *Curves and Surfaces in Geometric Modeling: Theory and Algorithms*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Glassner, A. (2004), Digital Cubism, *IEEE Computer Graphics and Applications* **24**(3), pp. 82-90.

Glassner, A. (2004), Digital Cubism, Part 2, *IEEE Computer Graphics and Applications* **24**(4), pp. 84-95.

Gobbetti, E. & Marton, F. (2005), Far Voxels: A Multiresolution Framework for Interactive Rendering of Huge Complex 3D Models on Commodity Graphics Platforms, *ACM Trans. Graph.* **24**(3), pp. 878-885.

Hauser, H. (2003), Generalizing Focus+Context Visualization, *in* Scientific Visualization: The Visual Extraction of Knowledge from Data (Proc. of the Dagstuhl 2003 Seminar on Scientific Visualization), Springer, pp. 305-327.

Hwa, L. M.; Duchaineau, M. A. & Joy, K. I. (2004),Adaptive 4-8 Texture Hierarchies, *in* VIS '04: Proceedings of the Conference on Visualization '04, IEEE Computer Society, Washington, DC, USA, pp. 219-226.

Keahey, A. (1998),The Generalized Detail-In-Context Problem, *in* INFOVIS '98: Proceedings of the 1998 IEEE Symposium on Information Visualization, IEEE Computer Society, Washington, DC, USA, pp. 44-51.

Kurzion, Y. & Yagel, R. (1997), Interactive Space Deformation with Hardware-Assisted Rendering, *IEEE Computer Graphics and Applications* **17**(5), pp. 66-77.

Leung, Y. K. & Apperley, M. D. (1994), A Review and Taxonomy of Distortion-Oriented Presentation Techniques, *ACM Transaction on Computer-Human Interaction* **1**(2), pp. 126-160.

Levoy, M. & Hanrahan, P. (1996), Light Field Rendering, *in* Proceedings ACM SIGGRAPH 1996, ACM, New York, NY, USA, pp. 31-42.

Li, Y.; Shum, H.; Tang, C. & Szeliski, R. (2004), Stereo Reconstruction from Multiperspective Panoramas, *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(1), pp. 45-62.

Lindstrom, P. & Pascucci, V. (2002), Terrain Simplification Simplified: A General Framework for View-Dependent Out-of-Core Visualization, *IEEE Transactions on Visualization and Computer Graphics* **8**(3), pp. 239-254.

Löffelmann, H. & Gröller, E. (1996), Ray Tracing with Extended Cameras, *Journal of Visualization and Computer Animation* **7**(4), pp. 211-227.

Mackinlay, J. D.; Card, S. K. & Robertson, G. G. (1990), Rapid Controlled Movement Through a Virtual 3D Workspace, *in* SIGGRAPH '90: Proceedings of the 17th Annual Conference on Computer graphics and Interactive Techniques, ACM, New York, USA, pp. 171-176.

Patterson, T. (2000), A View From on High: Heinrich Berann's Panoramas and Landscape Visualization Techniques For the US National Park Service, *Cartographic Perspectives* **36**, pp. 38-65.

Premoze, S. (2002), Computer Generated Panorama Maps, *in* Proceedings 3rd ICA Mountain Cartography Workshop. Mt. Hood, Oregon.

Roman, A.; Garg, G. & Levoy, M. (2004), Interactive Design of Multi-Perspective Images for Visualizing Urban Landscapes, *in* VIS '04: Proceedings of the conference on Visualization '04, IEEE Computer Society, Washington, DC, USA, pp. 537-544.

Sander, P. V. & Mitchell, J. L. (2006), Progressive Buffers: View-Dependent Geometry and Texture LOD Rendering, *in* SIGGRAPH '06: ACM SIGGRAPH 2006 Courses, ACM, New York, USA, pp. 1-18.

Spindler, M., Bubke, M., Germer, T. & Strothotte, T. (2006), Camera textures, *in* Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia, ACM, New York, USA, pp. 295-302.

Stone, M. C., Fishkin, K. & Bier, E. A. (1994), The movable filter as a user interface tool, *in* Proceedings of the Conference on Human Factors in Computing Systems, ACM, New York, USA, pp. 306-312.

Vallance S. & Calder, P. (2001), Multi-perspective images for visualization, *in* ACM International Conference Proceeding Series, Vol. 147, ACM, New York, USA, pp. 69-76.

Whitfield, P. (2005), Cities of the World. A History in Maps, The British Library, London.

Wonka, Peter; Wimmer, Michael & Francois Sillion (2001), Instant Visibility, *in* A. Chalmers & T.-M. Rhyne, ed.,Proceedings of Eurographics 2001, The Eurographics Association and Blackwell Publishers, pp. 411-421.

Yang, Y.; Chen, J. X. & Beheshti, M. (2005), Nonlinear Perspective Projections and Magic Lenses: 3D View Deformation, *IEEE Computer Graphics and Applications* **25**(1), pp. 76-84.

Yu, J. & McMillan, L. (2004), A Framework for Multiperspective Rendering, *in* Alexander Keller & Henrik Wann Jensen, ed., Rendering Techniques 2004, Proceedings of Eurographics Symposium on Rendering 2004, EUROGRAPHICS Association, pp. 61-68.

Zanella, A., Carpendale, M. S. T. & Rounding, M. (2002), On the effects of viewing cues in comprehending distortions, *in* Proceedings of the second Nordic conference on Human-computer interaction, ACM, New York, USA, pp. 119-128.