Computers & Geosciences 52 (2013) 389–397

Contents lists available at SciVerse ScienceDirect

Computers & Geosciences

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

Efficient occlusion-free visualization for navigation in mountainous areas

Hao Deng [a,b], Liqiang Zhang [b,*], Chunming Han [c], Yingchao Ren [c], Liang Zhang [b], Jonathan Li [d]

[a] School of Geosciences and Info-Physics, Central South University, Changsha 410083, China
[b] State Key Laboratory of Remote Sensing Science, Beijing Normal University, Beijing 100875, China
[c] Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China
[d] Department of Geography & Environmental Management, Faculty of Environment, University of Waterloo, 200 University Avenue West, Waterloo, ON, Canada N2L 3G1

ARTICLE INFO

Article history:
Received 11 June 2012
Received in revised form
9 September 2012
Accepted 21 September 2012
Available online 5 October 2012

Keywords:
Terrain deformation
Navigation
Non-perspective views
Visualization

ABSTRACT

In three-dimensional (3D) navigation, if mountainous terrain is displayed based on ordinary perspective projection, viewers often find that the features of interest are occluded, which prevents an overview of the features. This paper presents an approach for the automatic generation of consecutive non-perspective views of mountainous terrain. The proposed method can generate views without occlusions of important features, and allows viewers to navigate the landscape. The ray-tracing technique is employed to detect occlusions. The local elevations that occlude important features are transformed, while the resemblance and realism of the 3D landscape are maintained by solving global optimization problems. The approach maximizes the visibility of the features of interest on the deformed terrain. It also maintains a good balance between the elimination of occlusion and the preservation of resemblance. The occlusion-free visualization framework satisfies the demand for navigation and tour guidance in mountainous areas at interactive frame rates.

1. Introduction

Three-dimensional virtual landscapes have a wide range of applications in commercial systems and scientific research. Three-dimensional egocentric navigation or tour guide systems have several major advantages over those based on 2D maps (Chittaro and Burigat, 2005) for certain tasks. By egocentric, we mean that in navigation we observe the landscape from the "first person viewpoint" which is kept as close to ordinary pedestrian view as possible. So the egocentric views are more useful for the quick cognition of objects or shapes. Users can obtain geographic information efficiently and quickly so that they may concentrate more on tasks, such as way finding, tour planning and traveling.

However, in rugged mountainous areas or downtown areas with a myriad of buildings, the occlusion effect becomes a serious problem. Some features of interest (FOI), such as routes, can be out of the viewshed (Fig. 1). This often causes users to have difficulty in obtaining an overview of the important features and surroundings. For example, when a route being traveled disappears behind a mountain, a driver cannot predict and prepare for turns or crossroads on the occluded upcoming route. Although use of voice instructions can help to alert the driver, the instructions must be kept simple to avoid increasing cognition loads. Thus they cannot describe complex situations or provide sufficient indication well (Chittaro and Burigat, 2005). Therefore, occlusion-free visualization using non-perspective projection is an appropriate alternative for 3D navigation and tour guidance in mountainous areas.

The non-perspective terrain views can be comprehended as composite projections of 3D scenes from multiple viewpoints. In this way, these views can disclose occluded parts in ordinary perspective views. A satisfying non-perspective (or panoramic) view preserves the 2D visual effects similar to the perspective projection, while the visibility of important features is guaranteed. Generally, manually creating such an occlusion-free view for navigation and tour guidance is a tedious task and requires fair artistic skills. In recent years, automated techniques have emerged for occlusion reduction in geographical landscapes (Takahashi et al., 2006; Möser et al., 2008; Degener and Klein, 2009; Cui et al., 2010). These techniques provide powerful, non-perspective solutions for navigation and tour guidance. However, they can either introduce misleading distortions to the scenes or can be difficult to apply for visualization at satisfactory frame rates. For applications such as navigation, the minimized distortions facilitate the users' perception of their surroundings, and interactive visualization provides a better experience.

* Corresponding author. Fax: +86 10 58805274.
E-mail address: zihaozhang2003@yahoo.com.cn (L. Zhang).

0098-3004/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cageo.2012.09.016

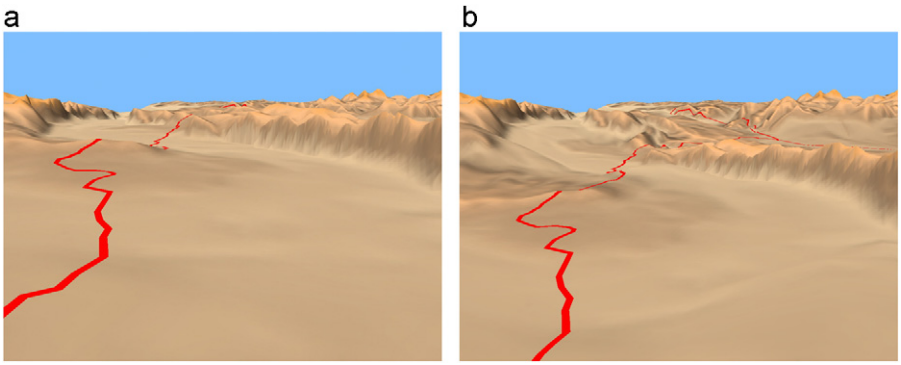

Fig. 1. Screenshots of an ordinary perspective view (a) and its occlusion-free view (b) generated by our approach.

This paper presents a framework for occlusion-free visualization of important features in mountainous areas that runs at interactive frame rates with low distortion. Our approach automatically deforms the terrains to generate non-perspective views according to the viewpoints, so that the occluded important features (e.g., roads, lakes and landmarks) can be revealed, and provides better views of the terrain to the users. Compared with the other methods, this framework avoids large distortions in highly occluded scenarios. Our main contributions to this field of visualization consist of (1) an efficient approach to detect occlusions and calculate the optimized rearrangement of the local geographical features, (2) a two-step framework to enable occlusion-free terrain visualization at interactive frame rates, and (3) a formulation of the terrain elevation transformation to minimize the deviation and discontinuities while avoiding occlusions.

2. Related work

The techniques of occlusion reduction have been studied for many years. Elmqvist and Tsigas (2008) provided a comprehensive overview and classification of the existing 50 methods, in which the techniques using dataset distortion, as they call volumetric probe in the literature, are more closely related to our work. In an earlier study, Elmqvist and Tudoreanu (2006) compared volumetric probe with simple navigation and fisheye transparency techniques, and found that volumetric probe is more stable for a wide range of task and for applications where mistakes incur a high cost. Carpendale et al. (1997) cleared line of sight to reveal sections previously obscured. McGuffin et al. (2003) used the deformation strategy for browsing inner portions of volumetric data. Degener et al. (2008) reduced the occlusion of short routes by aligning the route with the view direction. Qu et al. (2009) presented a "focus-context" zooming technique to broaden the occluded roads with other context being maintained. Popescu et al. (2010) and Cui et al. (2010) proposed camera models enabling interactive visualization, which achieve disocclusion by bending rays to circumvent occluders. But high levels of distortion introduced in the target surroundings are a major concern.

This paper presents a method for occlusion reduction used in geographical landscapes. In this field, panorama maps created by cartographers could provide a good overview and avoid occlusions of important features. Premoze (2002) developed a system allowing users to vertically scale and rotate the terrain. Takahashi et al. (2002) also presented an approach for generating panorama guide-maps, which can calculate the landscape displacement and rotation semi-automatically. Falk et al. (2007) used non-linear ray-tracing to render non-occluded views. These approaches, however, are not able to produce panoramic views automatically.

To the best of our knowledge, there are only a few methods reported in literature for automatically generating panoramic-like views. Premoze (2002) developed a system allowing users to vertically scale and rotate the terrain. Möser et al. (2008) developed interactive context-aware visualization systems that reveal features of interest by distorting the scene dramatically, although the distortion causes the resulting views to vary to an unacceptable extent from the ordinary perspective. Takahashi et al. (2006) proposed an occlusion-free route animation for car navigation, in which occlusion is eliminated by the arrangement of the landmarks onscreen. This process is tedious for generating occlusion-free views efficiently if the number of landmarks is large. Because the landmarks need to be sampled adequately to consider all occlusion situations, it is difficult use a small number of landmarks without introducing artifacts. Degener and Klein (2009) used non-linear optimization to search for terrain deformation to produce panoramic maps. Their approach considers two factors, resemblance and visibility, to constrain the possible deformation. Although the non-linear optimization process can obtain pleasing deformation results, its computational cost is too high for interactive applications. Deng et al. (2011) presented a method for generating panoramic map-like views in mountainous areas. The generated views integrate camera elevation and terrain deformation to delete the occusions. It introduces small distortions but do not preserve, or even largely deviate, the view height.

3. Overview

To avoid occlusions of FOI, we alter terrain shapes such that non-perspective effects are introduced for revealing the occluded features. Note that the method in Deng et al. (2011) introduces higher viewing positions to enhance visibilities of FOI. In contrast, our method preserves the original viewing positions that users favor, and only utilizes deformation strategy to disocclusion. Since no contribution of camera elevation in disocclusion would introduce higher distortion to the terrain, we hope to better pereserve terrain shape rather than simply scale down the occluding mountain top without explicitly considering terrain resemblance in Deng et al. (2011). The problem we solve can be demonstrated as follows.

Given the terrain T, a viewpoint P above the terrain surface and a set of FOI F on T, we obtain a view V in the standard perspective observed from P. Here we focus on generating a similar view V in which F is visible, while keeping V' similar to V by deforming T to T'. So what we solve is the deformed terrain T' whose deformation is tried to minimized. With T' solved, occlusion-free visualization of features of interest on the terrain is achieved. Here the terrain is represented as regular grid-based digital elevation models (DEMs).

Our method solve for the terrain using an analysis and deformation strategy. The analysis process is the same as Deng et al. (2011),

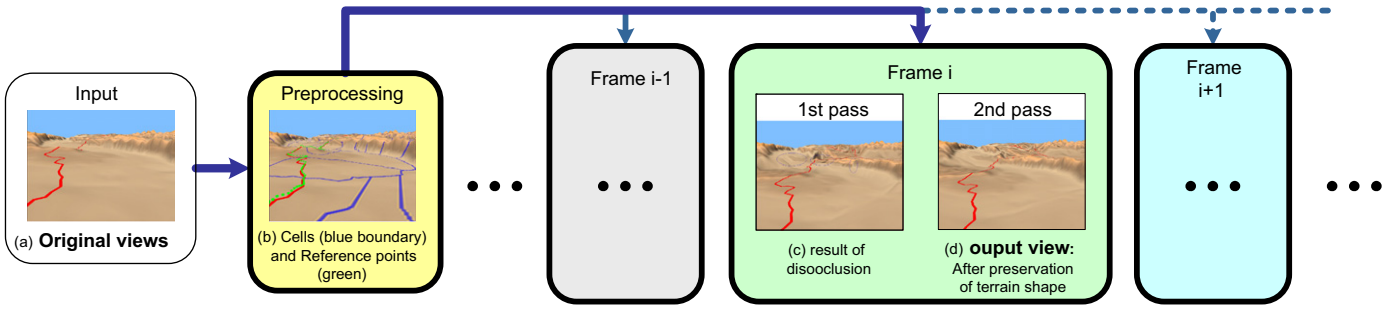

Fig. 2. Overview of our occlusion-free visualization.

which is done in preprocessing. Here we only briefly demonstrate this process. In analysis process we decompose the terrain into cells according to *Morse theory*, and set reference points on FOI to represent its visibilities (Fig. 2b). Each cell here corresponds to a mountain top, which is regarded as a potential occluder.

The deformation process is performed on a frame-by-frame basis. For each frame, we use two passes to solve for a shape-preserved deformed terrain without occluding FOI. First we deform the terrain locally by scaling down and push down the occluding cell to eliminate occlusion (Fig. 2c). And in next pass we perform a global deformation to preserve the shape of entire terrain (Fig. 2d). This two-pass framework enables solve the terrain at interactive frame rates. This is because during disocclusion pass, we only tackle these cells to eliminate occluders, which means disocclusion is solved in abstracted cell level. Disocclusion at this level is highly simplified compared to working on vertices level (Degener and Klein, 2009) or landmark level (Takahashi et al., 2006). The details of these two passes will be demonstrated in Sections 4 and 5, respectively.

4. Disocclusion

In this pass, we trace the line-of-sight (LoS) to the reference points. During each tracing, we detect the occluding cells, and then scale and push down the corresponding elevations to clear the LoS, such that the reference points become visible. Note that the tracing follows the order from background reference points to foreground points, please see Deng et al. (2011) for the details. The transformations are implicitly represented by the transformation parameters, which are organized by cells. Each cell has a set transformation parameters associated to elevation scaling and vertical displacement.

We use a greedy approach to compute the transformation parameters of the occluder regions. By greedy we mean that during the ray-tracing process, once any occlusion is detected, we compute the optimized transformation parameters for the local area to avoid occlusions (Fig. 3). Although this greedy approach does not produce an optimal solution for transformation of the entire terrain, this heuristic is robust and simple. It does a good job in balancing the avoidance of occlusion and the minimization of transformation. We leave for the optimal solution for the global terrain deformation in the next pass.

4.1. Transformation strategy

A reference point is invisible when (1) the reference point is occluded by the other parts of the terrain and (2) the reference point is located on the back-facing surface below the silhouette. Here we define the occlusion of situation (1) as regular occlusion, while the situation (2) as back-facing occlusion. And we tackle these two situations by employing two transformation strategies to the occluders (Fig. 4).

For a regular occlusion situation, only scaling and pushing down the cell are able to clear the LoS (Fig. 4a). For a back-facing occlusion, we transform the occluder part of the cell and leave the remaining part unaltered. Here, the occluder part is the region that is farther than the reference point from the view position in the 2D terrain domain (Fig. 4b).

When an occluding cell is identified, we should distinguish the occlusion between regular occlusion and back-facing occlusion. For a back-facing occlusion, the hit is the reference point itself; that is, the occluder and the occludee are located in the same cell. So conversely, if the occluder and the occludee are located in different cells, the occlusion is regular.

But if the hit and the reference point are located in the same cell, the occlusion can be both two situations. When the hit is the reference point itself or close to the reference point, the occlusion would be a back-facing occlusion. Otherwise, we still treat it as a back-facing occlusion by dividing the LoS into halves at the hit. Let us illustrate the case with a simple scenario. As shown in Fig. 5, assume that the reference point $\boldsymbol{P}$ and hit $\boldsymbol{Q}$ are located in the same cell. For the LoS $\boldsymbol{PV}$, because the line segment $\boldsymbol{PQ}$ before the hit $\boldsymbol{Q}$ is not blocked, we deal only with the other part of the LoS $\boldsymbol{QV}$ behind the hit. Note that the occlusion of $\boldsymbol{Q}$ is back-facing. Thus, we take $\boldsymbol{Q}$ as an additional reference point and treat the occlusion of $\boldsymbol{Q}$ as back-facing. This is performed by stopping the current ray tracing from $\boldsymbol{P}$ to $\boldsymbol{Q}$ and pushing $\boldsymbol{Q}$ into the priority queue of reference points for a new ray tracing from the hit $\boldsymbol{Q}$ in the following process.

4.2. Computation of the transformation parameters

As previously mentioned, the transformation is the combination of vertical scaling and displacement. The scaling-only transformation can be defined as

$$T_S(h) = sh, \tag{1}$$

where s is the scale factor, and h is the terrain elevation. The displacement-only transformation is defined as

$$T_t(h) = h + d, \tag{2}$$

where d is the displacement value. Combining T_S with T_t linearly, we obtain

$$T(h) = \lambda T_S(h) + (1-\lambda)T_t(h), \tag{3}$$

where λ is the internal ratio and is restricted to the interval [0, 1].

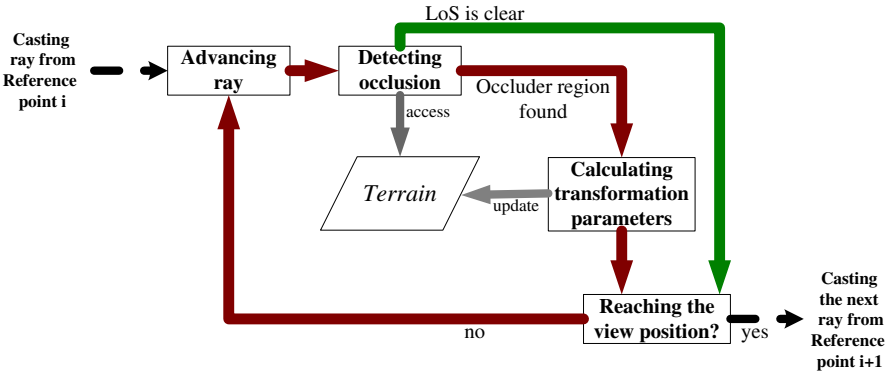

Fig. 3. A ray-tracing loop in which the terrain is dynamic.

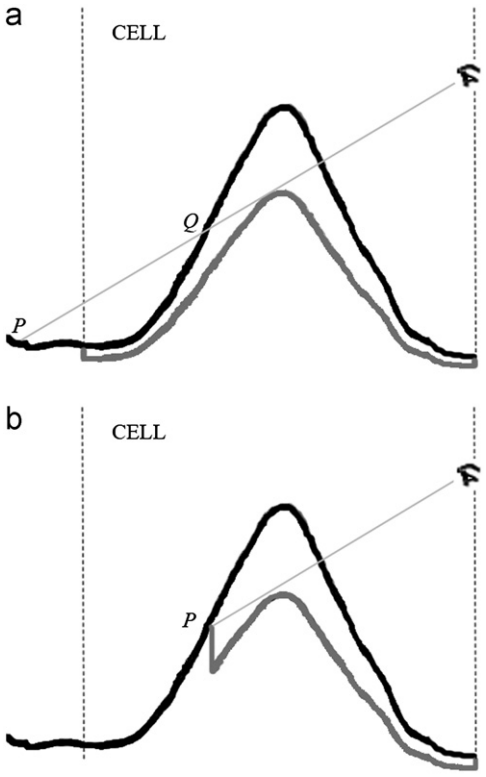

Fig. 4. Different occlusions and their corresponding transformations: (a) regular occlusion and (b) back-facing occlusion.

Finally, we obtain

$$T(h) = Sh + t, \tag{4}$$

where S is the scaling factor, and t is the displacement value.

To eliminate the occlusions, we consider the scaling-only transformation,

$$s^* = (h_h - \varepsilon_s)/h_0, \tag{5}$$

where h_h is the height of a hit, h_0 is the current height during ray tracing, and ε_s is the infinitesimal, which ensures that $T(h)$ is below the current LoS. In the same way, considering the displacement-only transformation only, we obtain

$$d^* = h_h - h_0 - \varepsilon_d, \tag{6}$$

where ε_d is also an infinitesimal, ensuring that h is below the current LoS. Combining the scaling-only and displacement-only transformations and using Eqs. (1)–(3), we obtain a transformation $T^*(h)$, which eliminates the occlusions, with the corresponding

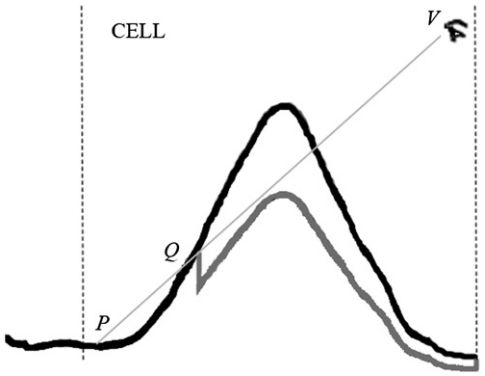

Fig. 5. The reference point and the hit located in the same cell. Point ***P***: the reference point. Point ***Q***: the hit. Dashed line: the boundary of the cell in the 2D scenario.

scaling factor S^* and displacement value t^*:

$$\begin{aligned} T^*(h) &= [(s^* - 1)\lambda + 1]h + (1-\lambda)d^* \\ &= S^*(\lambda)h + t^*(\lambda) \end{aligned} \tag{7}$$

Note that here T^*, S^* and t^* are also functions of λ. To find an unknown λ, we apply an optimization that combines two energy terms, corresponding with the deviation and the discontinuity that the transformation causes.

The definition of the deviation term is based on the following observation. In the part higher than the hit (*higher part*; the *lower part* can be defined in the same manner), the displacement introduces less deviation than scaling, and we should thus set the penalty term for the scaling operation to the higher part. Conversely, scaling introduces less deviation to the lower part, so the other penalty term is for the displacement operation on the lower part. We formulate the following energy term,

$$E_{\text{dev}}(\lambda) = \kappa \left| (1 - S^*(\lambda))h_h \right|^2 + (1-\kappa)t^{*2}(\lambda), \tag{8}$$

where h_h is the mean original elevation of the higher part, and κ is the percentage of the higher part grid vertices. In the implementation, both of these variants, corresponding to different hit elevations, can be pre-computed during watershed segmentation and be saved as a lookup table. We find that excessively reducing the terrain elevations often causes the terrain to be too greatly deformed. In practice, $\mu = 10.0$ is a proper value.

To define the discontinuity energy, we measure the difference in the scaling and displacement components between the target region and its neighbors set N by

$$E_{\text{dis}}(\lambda) = \sum_{r \in N} w_r \left| (S^*(\lambda) - S_r)\bar{h} \right|^2 + \mu \sum_{r \in N} w_r \left| t^*(\lambda) - t_r \right|^2, \tag{9}$$

where S_r and t_r are the associated scaling and displacement parameters of the neighbors, respectively, $\bar{h}$ is the mean original elevation of the target region, and w_r is the percentage of the boundaries with the neighboring region within the boundaries of the entire target region. We set the parameter μ to 10.0 because the displacement difference introduces higher discontinuities.

Combining Eqs. (8) and (9), we solve for the value of λ in Eq. (7) by minimizing

$$E(\lambda) = E_{\text{dev}}(\lambda) + \eta E_{\text{dis}}(\lambda), \tag{10}$$

where η is the parameter to balance these two terms. We found that setting η in the range from 1 to 10 works well, and the effect of η is shown in Section 6.

5. Terrain deformations

The previously described greedy approach calculates the optimized transformation parameters only for local regions. Using these transformation parameters on the elevation of the original terrain, we obtain only a naive deformed terrain of existing artifacts (highlighted in ellipses Fig. 2c). Although such a deformed terrain can succeed in eliminating the occlusions of important features, it still fails to maintain the global resemblance, such as in the surface details and continuities of the original terrain, because the regions are transformed by different parameters. To tackle this problem, we search for a terrain surface that preserves the terrain shape while maintaining the elevation of the naive deformed terrain and avoiding occlusions.

We maintain the terrain resemblance by preserving the Laplacian of the original terrain elevations (Sorkine et al., 2004), which approximate the local shape characteristics including the normal direction and mean curvature of the surface. Let a height function f represents the original terrain, and f'_0 represents its corresponding naive deformed terrain. The final deformed terrain f' is computed by solving the following variational minimization problem over the definition domain Ω of the terrain:

$$f' = \arg\min_{f'} \iint_{\Omega} |\Delta f' - \delta|^2 + w^2 (f' - f'_0)^2 dxdy, \tag{11}$$

where $\Delta(\cdot)$ is the Laplacian operator, and δ is the Laplacian of f. The solution of Eq. (11) produces a terrain surface with Laplacian coordinates that are closest to those of the original terrain, with the elevation similar to that of the naive deformed terrain. The effect of the weighting factor w on f' will be demonstrated in Section 6.

The discretization of Eq. (11) leads to an overdetermined linear system, which can be rewritten as $\boldsymbol{Ax}=\boldsymbol{b}$. We can obtain the final deformed terrain by solving the linear system in terms of the least squares:

$$\boldsymbol{A}^T\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}, \tag{12}$$

where $\boldsymbol{A}^T\boldsymbol{A}$ is a positive-definite sparse matrix, x is the vector denoted as x' that stacks the values of f'. Because the weights of the Laplacian operator and the weighting factor w in Eq. (11) in $\boldsymbol{A}$ is fixed, $\boldsymbol{A}^T\boldsymbol{A}$ is invariant for each frame. Therefore, we factorize only $\boldsymbol{A}^T\boldsymbol{A}$ and calculate the invariant component δ in $\boldsymbol{b}$ only once during preprocessing. In every frame, we simply perform the multiplication $\boldsymbol{A}^T\boldsymbol{b}$ and back-substitution to solve $\boldsymbol{x}$.

However, the back-substitution step is still too time-consuming for the interactive deformation because $\boldsymbol{A}^T\boldsymbol{A}$ may be too large for large terrain models. Therefore, hierarchical terrain models are employed to reduce the size of the linear system. We decompose f into two components with different resolutions corresponding to low and high frequencies. The low-frequency component $\check{f}$ is represented by the terrain with lower resolution. Thus, we can obtain a reduced linear system corresponding to this level. In every frame, the reduced linear system is first solved, and the final deformed terrain is then reconstructed by adding the scaled high-frequency details to the deformed terrain at the finest level. Through this process, we bypass solving a large linear system in every frame, and thus high deformation efficiency is gained.

To keep the controllability of the reduced model, we further add the elevation constraints at any position at fine levels. The elevation constraints are added in the cell where the important features are located. Because high mountains on the terrain are more likely to be deformed, the maximum height corresponding to each convex part is also set as the elevation constraint.

The elevation at any point $\boldsymbol{p}$ at fine levels can be represented by the linear combination of the elevations at four points enclosing it at the coarse level, e.g., using the bi-linear interpolation:

$$f_{\mathbf{p}} = \sum_{i}^{4} \varphi_i \check{f}_i, \tag{13}$$

where φ_i is the coefficients and satisfies $0 \le \varphi_i \le 1$. Then, we have the following elevation constraints:

$$\mathbf{W}\mathbf{x}' = \mathbf{h}, \tag{14}$$

where $\boldsymbol{W}$ is the matrix of coefficients, and h is the vector of additional elevation constraints.

To generate temporally coherent occlusion-free views, the temporal coherence of the final deformed terrains should be retained in the successive frames. We further add the temporal coherence constraint

$$\gamma \mathbf{I}\mathbf{x}' = \gamma \overleftarrow{\mathbf{x}}', \tag{15}$$

where $\overleftarrow{\mathbf{x}}'$ denotes the vector that stacks the elevations of the final deformed terrain in the last frame, and γ is the weight of this

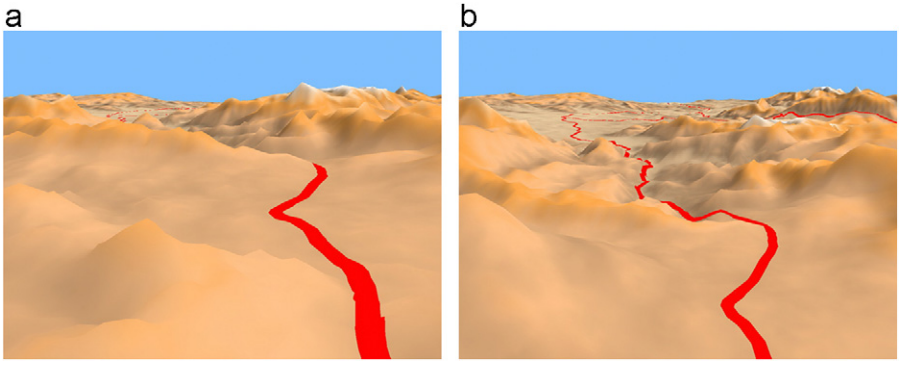

Fig. 6. Occlusion-free screenshots from our navigation system of the western Hunan region showing a downhill road: (a) a perspective view and (b) an occlusion-free view.

constraint and is set to $2.0 \times w$ in our navigation application. However, if the viewpoint changes too much between two successive frames (e.g., the viewpoint is switched by the user to a faraway position), causing the two frames to be temporally inconsecutive, we should not retain temporal coherence of the deformation. We can set $\overleftarrow{\mathbf{x}}$ to be the naive deformed terrain such that the final deformed terrain in the current frame is not influenced by the deformation of the previous frame.

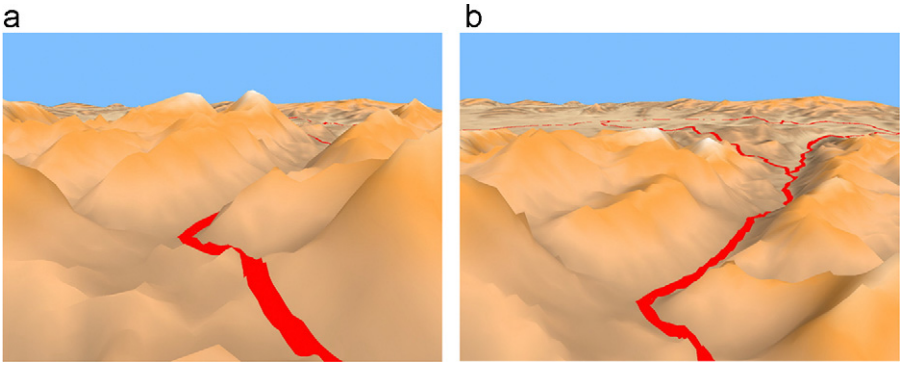

Fig. 7. A route in the valley in the western Hunan region: (a) a perspective view and (b) an occlusion-free view.

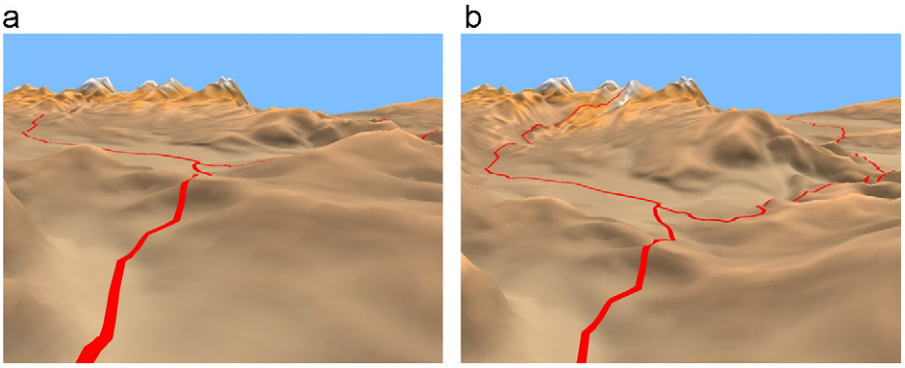

Fig. 8. Hiking trails in the northern Shanxi region. (a) perspective view, (b) occlusion-free view.

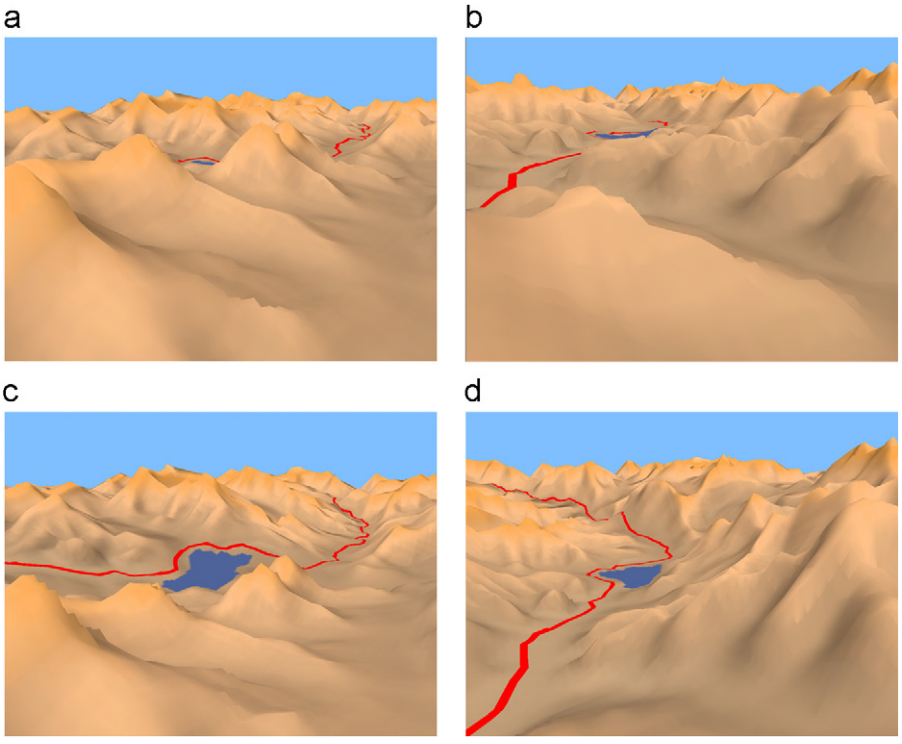

Fig. 9. Screenshots from our navigation system of a reservoir and a road in the northern Shanxi region. (a) and (b) show ordinary perspective projections, whereas (c) and (d) are their respective corresponding occlusion-free views.

Combining the constraints, we obtain the final matrix and vectors for the overdetermined linear system $\boldsymbol{Ax}=\boldsymbol{b}$:

$$A=\begin{pmatrix}\mathbf{L}\\ w\mathbf{I}\\ \mathbf{W}\\ \gamma\mathbf{I}\end{pmatrix},\quad b=\begin{pmatrix}\delta\\ w\mathbf{x}_0'\\ \mathbf{h}\\ \gamma\overleftarrow{\mathbf{x}}'\end{pmatrix}\quad\text{and}\quad x=\mathbf{x}' \tag{16}$$

where $\mathbf{L}$ is the matrix containing the weights of the Laplacian operator, $\mathbf{I}$ is the identity matrix, and $\mathbf{x}_0'$ is the vector that stacks the values of the naive deformed terrain. Note that the matrix $\boldsymbol{A}$ here is still invariant and sparse, and thus does not hinder reusing factorization.

6. Results and discussion

To validate our method, two real scenes were modeled to show different configurations of terrain fields. Both of the scenes are in mountainous areas at the resolution of 1025×1025 vertices. The experiments were conducted on a PC with a Duo 2.33 GHz CPU, a 2 GB main memory, and a GeForce 9600 graphics card. Our accompanying video is available at the website http://www.youtube.com/watch?v=CZFl2uNgj6o.

With the ordinary perspective views, the terrains were rendered at 90 to 120 frames per second (fps). The occlusions-free views run at frame rates from 25 to 30 fps. The generated occlusion-free views are temporal coherent.

Here, we report only the CPU cost. A direct solver (Toledo 2003) was used to solve the linear system described in Section 5. During preprocessing, it required 0.14 s to build the hierarchal terrain model and construct the sparse matrix $\boldsymbol{A}^T\boldsymbol{A}$ with the size of $16{,}641\times16{,}641$. The solver required approximately 0.5 s to factorize $\mathbf{A}^T\mathbf{A}$. In each frame, the computation time to solve the linear system was approximately 0.03 s on the PC.

Two routes on the rugged terrain located in western Hunan Province, China were used to test the effectiveness of our

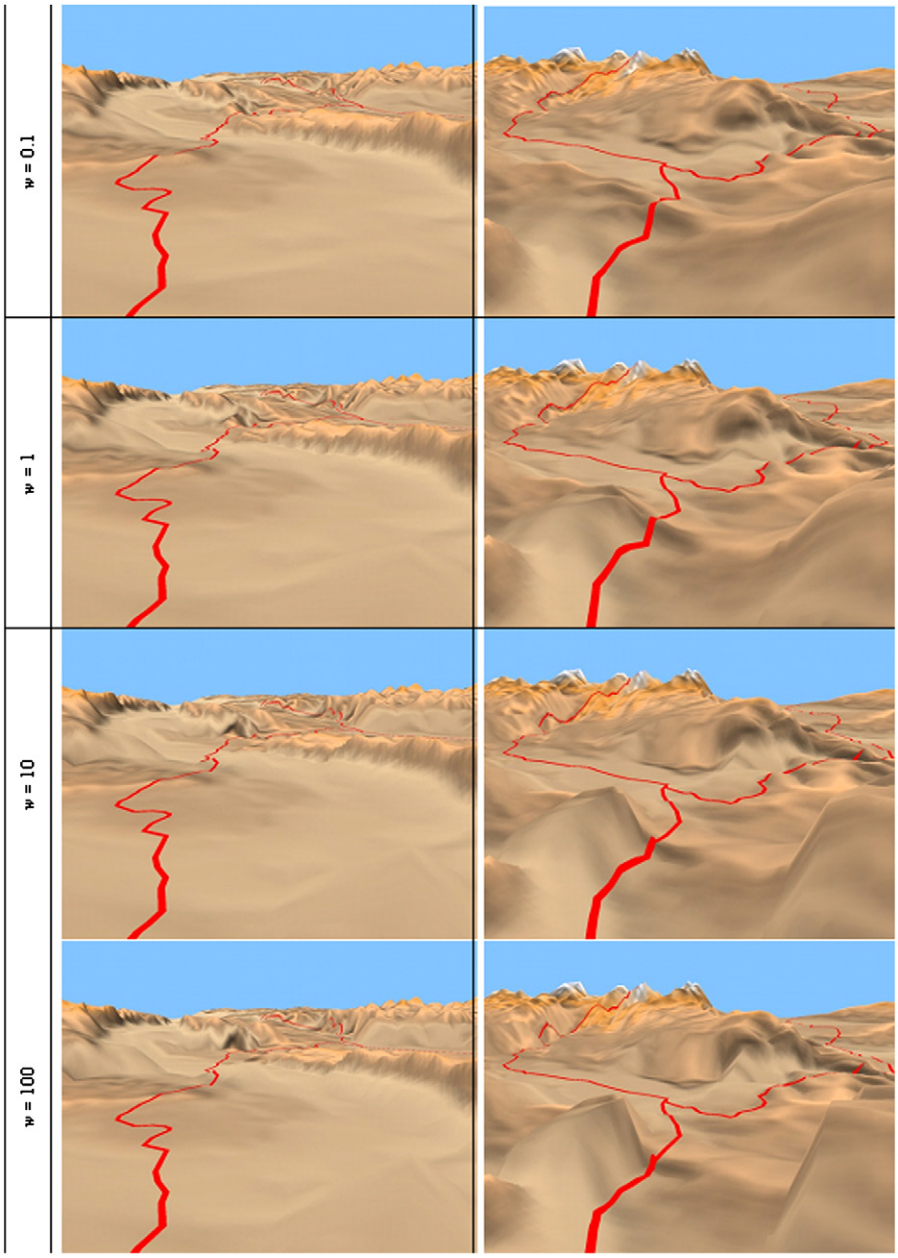

Fig. 10. Effects of w; each row corresponds with the result for a value of w.

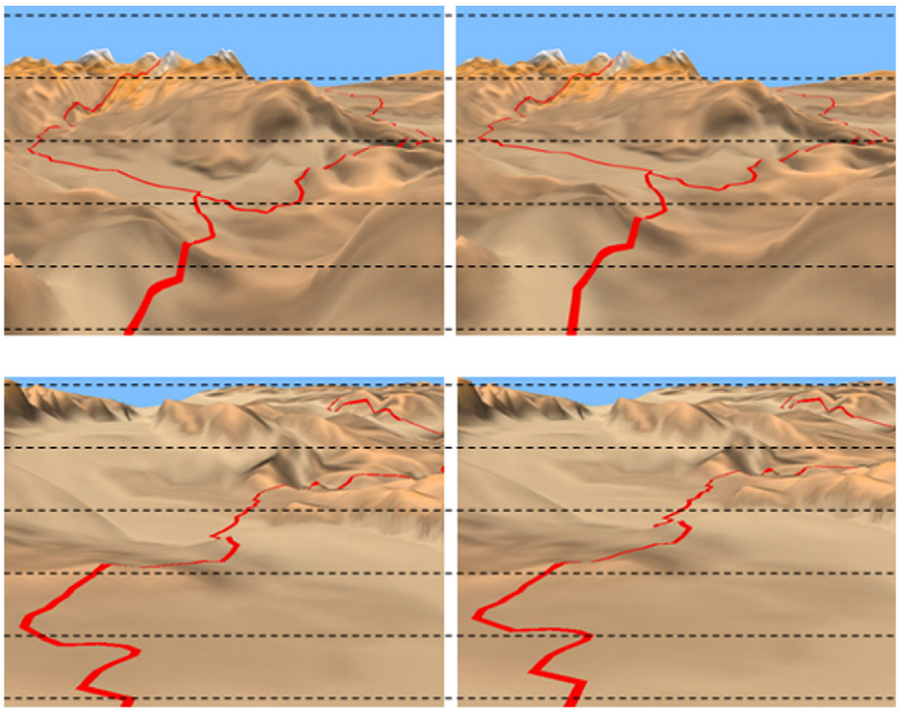

Fig. 11. The effect of η: the left and right columns show the result obtained by setting η to 0.1 and 10.0, respectively. The bottom row corresponds to a local part of Fig. 1. The dashed lines are added to facilitate the comparison.

Table 1
Times and reference points for generating non-perspective views.

	Fig. 1 (b)	Fig. 6 (b)	Fig. 7 (b)	Fig. 8 (b)	Fig. 9 (c)	Fig. 9 (d)
Number of reference points	513	670	660	560	195	202
Times (ms)	12	14	14	14	2	3

approach for navigation in mountainous areas. A total of 651 cells were extracted on the terrain during preprocessing. The partitioning process took 1.57 s. As shown in Figs. 6(b) and 7(b), the elevations of the terrain are transformed appropriately to reveal the route, while each convex part is preserved. These occlusion-free views provide viewers a good overview of the routes, while the closed egocentric perspective is maintained.

Figs. 1 and 8 illustrate the results applied to the hiking trails in the northern Shanxi scene, which are more twisted in the mountainous areas. In Fig. 8, the occluded trail on the left-hand side lies on the back-facing surface, while the LoS to the occluded trail on the right-hand side is blocked by the bumped mountains. However, our approach works well, and both of the trails are disclosed with preservation of the terrain details.

Fig. 9 exhibits additional examples, consisting of non-perspective images highlighting a road and a reservoir in northern Shanxi. Our approach succeeds in revealing the full view of the polygon feature while also revealing the road bypassing the reservoir (Fig. 9c and d).

Note that w in Eq. (11) is a factor that balances the terrain shape preservation and visibility of the important features. To demonstrate the effect of w on the final deformation, we used different values of w for the scenarios in which the features are highly occluded, as shown in Figs. 1 and 8. Fig. 10 illustrates the results when w increases from 0.1 to 100.0. It is reasonable that the discontinuities on the terrain would become more noticeable

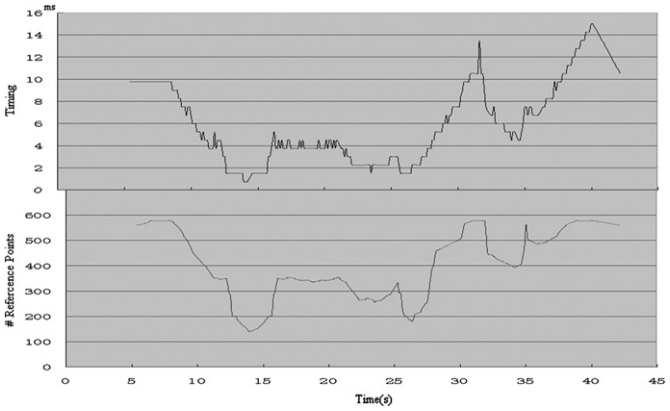

Fig. 12. Reference points vs. times of the LoS tracing and calculation of the transformation parameters.

as w increases, while the visibility of the important features is enhanced. Note that most of the important features are visible even when w is small. We find that setting w in the interval between 0.1 and 1.0 leads to a satisfactory result, which maintains an appropriate compromise between the shape preservation and visibility. In the experiments, $w=0.6$.

Fig. 11 illustrates the effect of the η parameters ($\eta=0.0$ and $\eta=10.0$, respectively) in Eq. (10) on the final deformation. Because of the order of the ray tracing, the discontinuities in the transformation are propagated from the distant regions to the near ones. As shown in Fig. 11, the distortions near the viewpoint are obviously reduced when the discontinuity term is considered. In practice, this approach works well when w is in the interval between 1.0 and 10.0 and when the final deformations are not especially sensitive to the value of η.

Because the transformation parameters were calculated by tracing the LoS only from the reference points in the view

frustum, the number of reference points varies in each frame. Table 1 lists the reference points and times of the LoS tracing and the calculation of the transformation parameters corresponding to Figs. 1, 6–9.

We have further compared the relationship between the reference points and the time cost by computing the transformation parameters. We navigate the terrain scene of the western Hunan region with occlusion-free views for a period of time, during which the numbers of reference points and times for ray tracing are recorded (Fig. 12). In the bottom diagram of Fig. 12, the number of reference points is kept small because of the frustum culling. In the upper diagrams, the times for calculating transformation parameters are always kept in real-time during navigation. These suggest that our greedy approach is efficient in finding the optimized transformation parameters.

The performance of our method maintains a low dependence on the graphics hardware, which facilitates the migration of our method to mobile platforms. Today, mobile phone CPUs with primary frequencies higher than 1 GHz become very common and have exhibited powerful capabilities. This rapid progress will enable an acceptable performance of our method on mobile platforms. Further scalability can be gained by only disocclusion of a section of the features in front of the user, and by deformation of terrain at a further reduced hierarchy.

7. Conclusions

This paper has presented an effective approach to the interactive 3D terrain visualization of mountainous areas. The occlusion-free views are effective and efficient in eliminating the occlusions of important terrain features caused by rugged mountains while maintaining the resemblance and realism of the 3D environment. Compared with other methods, our main contributions consist of (1) developing an approach that can detect occlusions and calculate the optimized local transformation in real-time and (2) presenting a two-step occlusion-free visualization framework, in which the optimized transformation parameters are first computed in local regions and are then used to search a global optimization deformation for rendering. The occlusion-free visualization of features of interest can be achieved at interactive frame rates.

Our future work will also include the use of non-linear optimization to establish a more general deformation in panoramic views. Moreover, additional appropriate occlusion management techniques (e.g., optimized viewpoint selection and transparency) will be applied to improve the reliability of the occlusion-free scene visualization.

Acknowledgments

The authors would like to thank the editors and the anoymous reviewers for their valuable comments and suggestions which have helped us improve the context and the presentation of the paper. The study is supported by the National Natural Science Foundation of China (No. 60502008) and the National 863 High-Tech Program of China (No. 2011AA120302 and 2012AA12A204).

References

Carpendale, M.S.T., Cowperthwaite, D.J., Fracchia, F.D., 1997. Extending distortion viewing from 2D to 3D. IEEE Computer Graphics and Applications 17 (4), 2–51.
Chittaro, L., Burigat, S., 2005. Augmenting audio messages with visual directions in mobile guides: an evaluation of three approaches. In Mobile HCI 2005, 107–114.
Cui, J., Popescu, V., Rosen, P., Hoffman, C., 2010. A curved ray camera for handling occlusions through continuous multiperspective visualization. Transactions on Visualization and Computer Graphics 16 (6), 1235–1242.
Deng, H., Zhang, L., Ma, J., Kang, Z., 2011. Interactive panoramic map-like views for 3D mountain navigation. Computers & Geosciences 37 (11), 1816–1824.
Degener, P., Klein, R., 2009. A variational approach for automatic generation of panoramic maps. ACM Transactions on Graphics 28 (1), 1–14.
Degener, P., Schnabel, R., Schwartz, C., Klein, R., 2008. Effective visualization of short routes. IEEE Transactions on Visualization and Computer Graphics 14 (6), 1452–1458.
Elmqvist, N. and Tudoreanu, M., 2006. Evaluating the effectiveness of occlusion reduction techniques for 3D virtual environments. In Proceedings of the ACM Symposium on Virtual Reality Software and Technology, 9-18.
Elmqvist, N., Tsigas, P., 2008. A taxonomy of 3D occlusion management for visualization. IEEE Transactions on Visualization and Computer Graphics 14 (5), 1095–1109.
Falk, M., Schafhitzel, T., Weiskopf, D. and Ertl, T., 2007, Panorama maps with non-linear ray tracing. In Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia (GRAPHITE'07). ACM, New York, pp. 9–16.
McGuffin, M.J., Tancau, L. and Balakrishnan, R., 2003, Using deformations for browsing volumetric data. Proc. IEEE Conf. Visualization, 401–408.
Möser, S., Degener, P., Wahl, R., Klein, R., 2008. Context aware terrain visualization for wayfinding and navigation. Computer Graphics Forum 27 (7), 1853–1860.
Premoze, S., 2002. Computer generated panorama maps. In Proceedings of the ICA Mountain Cartography Workshop.
Popescu, V., Rosen, P., Adamo-Villani, N. 2010. The graph camera. International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH Asia, 2009.
Qu, H., Wang, H., Cui, W., Wu, Y., Chan, M., 2009. Focus+context route zooming and information overlay in 3D urban environments. IEEE Transactions on Visualization and Computer Graphics 15 (6), 1547–1553.
Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M.C., Rössl, R.O., Seidel, H.-P., 2004. Laplacian surface editing. In Proceedings of the symposium on Geometry processing, pp. 175–184.
Takahashi, S., Ohta, N., Nakamura, H., Takeshima, Y., Fujishiro, I., 2002. Modeling surperspective projection of landscapes for geographical guide-map generation. Computer Graphics Forum 21 (3), 259–268.
Takahashi, S., Yoshida, K., Shimada, K., Nishita, T., 2006. Occlusion-free animation of driving routes for car navigation systems. IEEE Transactions on Visualization and Computer Graphics 12 (5), 1141–1148.
Toledo, S., 2003. TAUCS: A Library of Sparse Linear Solvers. version 2.2, Sept. 2003. Available online at 〈http://www.tau.ac.il/~stoledo/taucs/〉.