Immersive Urban Analytics through Exploded Views

Zhutian Chen*
Hong Kong University of Science and Technology

Huamin Qu†
Hong Kong University of Science and Technology

Yingcai Wu‡
Zhejiang University

ABSTRACT

Recent years have witnessed the rapid development and wide adoption of immersive head-mounted devices, such as HTC VIVE, Oculus Rift, and Microsoft HoloLens. These immersive devices have native support for displaying 3D models in an immersive environment. The capability has the potential to significantly extend the methodology of urban visual analytics by providing critical 3D context information and creating a sense of presence. However, 3D city models may cause severe occlusion problems, blocking the path toward enabling immersive urban analytics. This paper presents a novel technique to create an exploded view in immersive environments. The 3D models are firstly exploded and grouped into layers. Then we formulate and solve a nonlinear least squares optimization problem to ensure the visibility, compactness, and alignment of all layers. Different from existing methods, our method is tailored to analyze urban data in immersive environments. We further conduct experiments to compare our method with two different methods and exam our method in various situations based on the HoloLens platform.

Index Terms: Immersive Analytics; Urban Visual Analytics; Data Visualization; Exploded Views; Occlusion Free Visualization

1 INTRODUCTION

Urban visual analytics has been proven useful in solving various problems of urban cities, such as location selection [13], urban planning [9], and traffic analysis [8], by integrating the computational power of machines and the domain knowledge of experts. Most existing studies of urban visual analytics utilize 2D maps [24] on which every point is viewed overhead. As 2D maps create an abstraction of the real world, the maps lose significant context information on the urban environment, consequently leading to the severe limitations in solving space-related problems in the urban context. First, the lack of depth information of vertical cities poses a significant challenge for making informed decisions in many scenarios. Second, a 2D map that lacks the appearance of the real world cannot provide users with a sense of presence. Therefore, growing interest has been observed in applying 3D maps with real-world 3D models for urban visual analytics [6, 16].

In recent years, various of immersive head-mounted devices, such as HTC VIVE, Oculus Rift, and Microsoft HoloLens, have been invented and adopted in a wide range of settings. The immersive devices use stereoscopic techniques to create a natural support for 3D display, thereby creating an engaging and immersive visual environment [2]. The significant development and broad adoption of the immersive devices shed new light on visualizing heterogeneous geospatial urban data in such an immersive environment; this field can be referred to as *immersive urban analytics*.

Recently, based on Microsoft HoloLens, HoloMaps [20] has been introduced to display real-time urban data. Examples include traffic flow and tweets with geo-tags on Microsoft Bing Maps 3D to allow users to explore a city in an immersive environment. However, HoloMaps directly overlays the urban data on a 3D map. The occlusion problem and limited user interactions supported by HoloMaps make this application unsuitable for immersive urban analytics. However, there is a serious challenge lies in enabling immersive urban analytics.

3D city models can suffer from occlusion and result in a cluttered display. Modern cities are generally crowded with buildings occluding each other and nearby roads. In addition, users can be easily overwhelmed by abundant urban information, such as traffic flow, points of interest (POI), road network, and 3D buildings. Existing commercial systems [5, 7, 15] which can support 3D maps fail to address this problem properly. Consequently, users must rotate the 3D maps continuously to gain an overview and observe the occluded information. Thus, using these systems in an immersive environment would be tedious and prone to errors.

To address this issue, we present a novel method to generate an exploded view for 3D city models. In an exploded view, an object is separated (or "exploded") into several components to simultaneously convey the global structure of the depicted object and the local details of the components [3, 12]. Our method automatically explodes 3D city models driven by data and users' viewpoints, thereby displaying the urban data without occlusion while preserving users' mental map of the city context. We employ a tailored energy function to calculate the energy level of components and group them into different layers. Each layer must satisfy certain constraints that secure visibility, compactness, and alignment.

2 RELATED WORKS

2.1 Immersive Analytics

In recent years, with the popularization of low-cost immersive devices such as Leap Motion, Kinect, Oculus Rift and Microsoft HoloLens, an increasing number of researchers are developing an interest in Immersive Analytics, which aims to allow users to immerse themselves in data in a way that supports real-world analytical tasks [2] [18].

Many researchers have conducted studies on HMD-based immersive analytics. Kwon et al. presented a study on layout, rendering, and interaction methods for graph visualization in an virtual reality (VR) environment created by head-mounted display (HMD) [11]. The result showed that using their techniques is more highly efficient than traditional 2D graph visualization in HMD VR environments. Cordeil et al. developed a visually immersive prototype for collaborative aircraft trajectory visualization based on HMD VR devices [4]. Moran et al. attempted to study the geographic distributions of Twitter data by developing a 3D application utilizing an HMD VR setting [14]. However, they only displayed tweets with geo-tags on a 3D map. Users can explore those tweets through simple interactions. Tong et al. presented GlyphLens [22], a view-dependent occlusion management for interactive glyph visualizations, and explored the interaction of their system in VR headset.

Most of the existing studies focus on abstract data, especially the graph data. Few researchers in the field of data visualization have conducted explorations on urban data in immersive environments. To the best of our knowledge, our work is the first step in the direction of utilizing immersive technologies in urban visual analytics.

2.2 Exploded Views

Exploded views have been widely used in the industry to illustrate mechanical assemblies, considering its capacity to provide occlusion-free visualization to reveal the hidden detail of 3D objects.

*e-mail:zhutian.chen@ust.hk
†e-mail:huamin@cse.ust.hk
‡e-mail:ycwu@zju.edu.cn

Many computer graphic researchers have conducted research on generating exploded views. Agrawala et al. [1] presented a method that analyzed the blocking of information among different parts to generate an explosion graph. Based on the explosion graph, their system can automatically decide the explosion order and direction for each part without violating blocking constraints. Li et al. [12] extended Agrawala's work by considering part hierarchies of the CAD models and handling common cases of interlocking parts. Tatzgern et al. [21] proposed a method to reduce the complexity of an exploded view. Their method identifies similar objects and only one representative is exploded.

Apart from CAD models, some researchers have investigated methods to create an exploded view for various types of data. Bruckner and Gröller [3] presented a force-based exploded views for volume data. They modeled the relationships among various parts of the volume through physical forces. By adding repelling forces the occluding parts can be pushed away. Karpenko et al. [10] presented a technique that automatically generates an exploded view for a complex mathematical surface by partitioning it into parallel slices.

Most of these techniques focus on revealing two types of information regarding 3D objects, namely the internal structures of models and spatial relationships among various parts, whereas neither of them is the focus of urban visual analytics. Moreover, none of these techniques consider data during the views generation. Thus, these techniques cannot be simply applied to visual analytics in the urban context.

3 Method

Several illustration techniques can be utilized, such as cutaways, transparency, and deformation, to solve the occlusion problem. However, these methods remove, de-emphasize, or distort occluding geometries. In the urban context, retaining the context, including the information of occluders, is instructive. Therefore, we selected exploded views to handle the occlusion problem, which simultaneously conveys the global structure and the local detail of the context without distortion [3, 12].

In this section, we first describe a list of principles that an exploded view should follow in the context of immersive urban analytics. Then, we introduce our method gradually.

3.1 Principles

Li et al. [12] summarized some general principles for creating exploded views. However, given that their method is designed for mechanical assembly, some principles are inappropriate in our situation, such as canonical explosion directions, part hierarchy, and splitting containers. To create an effective and aesthetic exploded view for immersive urban analytics, we first attempt to answer the question of *"what components should be exploded"*, followed by the question of *"how components should be exploded."* Based on the two issues, we clarify some principles from Li's work and extend these principles in terms of immersive urban analytics. These principles can be organized into two parts based on the type of questions they address, namely *what* and *how*.

What to be exploded. Given that the *what* query is a *"from 0 to 1"* question, we address this question by starting from defining some atomic components. For simplicity's sake, in a 3D city model, we define two kinds of components, namely *building* and *region*. A building is represented by a 3D model, the primary source of occlusion. A region is a part of the ground embedded with roads. Several methods can be used to partition the ground. In our implementation, we adopted the DBSCAN [23] method, which is a data clustering algorithm, to partition the ground into regions based on data points, including traffic, mobile phone, and POI data. When creating exploded views, whether a component should be exploded or not should consider the following factors:

- W1 **Data Driven.** Avoiding excess visual clutter that only particular components, rather than all components, should be exploded is a key consideration.

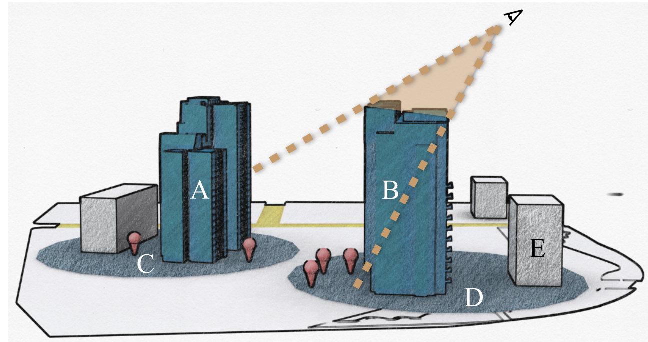

Figure 1: Energy of region D is higher than that of region C because the former contains more data points than the later; the energy of building B is higher than that of building A because the former occludes more data points and blocks a region with higher energy than the latter.

- W2 **Blocking Constrained.** Components blocked by others should not be exploded until the blocking parts are exploded.
- W3 **Wholeness Preserved.** A building component should be exploded if most of its neighbors are exploded, thereby providing a sense of wholeness and reducing the number of fragments.

How to be exploded. After acquiring the exploded components, we should group these components into layers by satisfying the following principles:

- H1 **Stratification.** Components should be grouped into layers to help users understand the overwhelming information well.
- H2 **Visibility.** The offset between layers should be set properly to eliminate the occlusions among layers.
- H3 **Compactness.** The distance between layers should be minimized to reduce the empty space in the view and to help users spontaneously connect the different layers.
- H4 **Alignment.** The parallel misalignment between a layer and its original position should be minimized for the reconstruction of the city to be mentally easy for users.

Satisfying the aforementioned principles is a great challenge. Existing methods [1, 3, 10, 12, 21] cannot achieve all of these principles simultaneously. In our algorithm, we follow three steps, namely, **transition**, **intra-layer**, and **inter-layer** optimization. We introduce these steps separately in the following sections.

3.2 Transition

The first step int our algorithm is determining which layer a component should be exploded to. We transferred the aforementioned principles of *what* into different energy terms. When a user requires exploding the 3D city model, we calculates the energy of each component based on these energy terms. If a component has added energy, it jumps to a relatively high layer. We defined each energy term in detail as follows:

Data Driven In the context of immersive urban analytics, only the components related to interesting information should be exploded (**W1**). The interesting information is defined by users, which could be POIs, landmarks, roads, and areas. This principle is twofold. First, the components occluding interesting data should be exploded to eliminate the occlusion. In Fig. 1, building B is an exploded candidate but building E is not, because the former occludes three POIs but the latter does not. Second, as the exploded components can draw the attentions of users, the components containing interesting information should be exploded to simplify the analysis. For example, in Fig. 1, given that region D contains three POIs which are the analyzed target of users, this region should be exploded to stand out from the background. To reflect this principle, we define the energy term of $component_i$ as $E_{di} = |O_{v,i}| + |D_i|$, where $O_{v,i}$ is the set of interesting information occluded by $component_i$ from the viewpoint v of the user, and D_i represents the set of interesting objects embedded in $component_i$. This energy term aims to ensure that the components occluding or containing interesting information have high energy. For example, in Fig. 1, region D, which contains three POIs, has higher energy than region C, which contains two POIs; building B, which occludes three POIs has higher energy than building A, which occludes one POI. In this study, the energy term only reflects the size of D_i and $O_{v,i}$. Other well-established energy term can also be used, such as information entropy.

Blocking Constrained If the blocking constraints are violated, then the exploded views would be counter-intuitive (W2). For example, users are confused when region C pass through building A and are exploded to a high level. To guarantee the blocking constraints, we define the energy term as $E_{bi} = \sum_{j \in B_i}(E_{dj} + E_{bj})$, where B_i is the set of components blocked by $component_i$, and E_{dj} and E_{bj} are the *data driven* energy and *blocking constrained* energy of $component_j$, respectively. For example, in Fig. 1, building A blocks region C; thus, building A inherits energy from region C. With this recursive energy term, we can assure that the energy of a component is always higher than that of the components that the component blocks.

Wholeness Preserved The adjacent buildings should be exploded to the same layer to retain the wholeness and reduce the number of fragments (W3). In traditional generation methods for exploded views, users require manually inputting the grouping relationship of components. This process is strenuous for users in the context of a 3D city model because of the substantial number of buildings. We adopt a heuristic method based on Gaussian smoothing, because the concept of wholeness is challenging to formalize. Other well-established methods can also be adopted. The energy term for building components is defined as $\forall i \in C_b, E_{wi} = \sum_{j \neq i \wedge j \in C_b} \frac{(E_{dj}+E_{bj})}{\sigma\sqrt{2\pi}} e^{-\frac{dist(i,j)^2}{2\sigma^2}}$, where C_b represents the set of building components, $dist(i, j)$ is the distance between $component_i$ and $component_j$, and σ is the smoothing radius. This energy term smoothens the energy of each building to its neighbors. For example, in Fig. 1, building E is shared with some energy by building B.

In summary, we use a weighted summation of the aforementioned energy term to define an energy function for calculating the energy of each component:

$$\forall i \in C, E_i = \omega_d E_{di} + \omega_b E_{bi} + \omega_w E_{wi}, \tag{1}$$

where ω_d, ω_b, and ω_w are the weighted parameters. After obtaining the energy, the algorithm divides components into different layers according to thresholds (H1):

$$L_i = \begin{cases} 0 & 0 \le E_i \le T_1 \\ 1 & T_1 < E_i \le T_2 \\ \cdots & \cdots \\ n-1 & T_{n-1} < E_i \le T_n \end{cases}, \tag{2}$$

where L_i is the target layer of $component_i$, and $T_0, T_1, \cdots T_n$ are the thresholds. In our implementation, we define two energy thresholds to divide the components into three layers. The first layer contains unexploded components on the 3D map; the second layer consists of regions and roads; the third layer consists of buildings. Our energy function only determines the layer to which a component should be exploded. The precise location of each component is calculated and identified in the following steps.

3.3 Intra-layer Optimization

After the first step, components are organized into layers. The second step of our algorithm is to optimize the components within each layer. Some deformations can be used in the 3D city models to achieve clarity and informativeness of a 3D map. For example, buildings can be grouped tightly; the ratios between the roads and buildings can be re-scaled to broaden the roads. Our method is pluggable and flexible. Any well-established techniques for optimizing the map can be utilized, such as [17] and [19]. In our implementation, we use Sun's method [19] to broaden the roads in an exploded region.

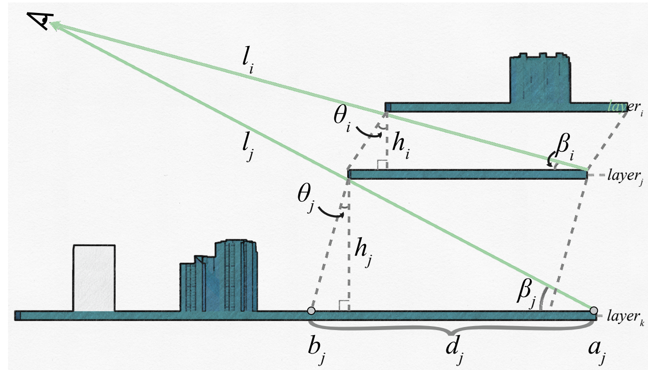

Figure 2: Illustration of the inter-layer optimization process. Suppose the $layer_j$ is being optimized. The vertical distance between $layer_j$ and $layer_k$ is denoted by h_j, which is minimized to ensure the compactness. We also minimized θ_j to retain the parallel alignment between $layer_j$ and its original position. To guarantee the visibility of $layer_k$, $layer_j$ must be placed upon the green line l_j.

3.4 Inter-layer Optimization

After the second step, the geometry of each layer is fixed because the size and relative position of each component are fixed. In the last step, our algorithm conducts an inter-layer optimization to satisfy the principles for layers by formulating the problem as a nonlinear least squares optimization problem. The layers are constrained to move parallel to the gaze direction to unfold the layers naturally and minimize the effort of users to follow the explosion. Then, the problem is reduced to a 2D one (Fig. 2). Without loss of generality, we assume that $layer_j$ is being optimized. To ensure the visibility (H2), $layer_j$ must be distributed not lower than the line l_j, which is defined by the position of the eyes of the user and the farthest point a_j of $layer_k$ in the gaze direction. We use the vertical distance h_j between $layer_j$ and $layer_k$ to reflect the compactness (H3). Given that h_j should be minimized, $layer_j$ must contact l_j. The parallel misalignment between $layer_j$ and $layer_k$ is depicted by θ_j, which should be minimal (H4). If b_j denotes the nearest point of the original position of $layer_j$ in the gaze direction, then the distance between b_j and a_j is d_j.We further denote the angle between l_j and $layer_k$ by β_j, which range is $\left(0, \frac{\pi}{2}\right)$. In Fig. 2, the following equation is computed easily:

$$h_j = \begin{cases} d_j \tan\beta_j & \theta_j = 0 \\ \frac{d_j \sin\theta_j \tan\beta_j}{\cot\theta_j + \tan\beta_j} & 0 < \theta_j < \frac{\pi}{2} \\ 0 & \theta_j = \frac{\pi}{2} \end{cases}. \tag{3}$$

In summary, for $layer_j$, we define an objective function as the summation of the square of normalized θ_j and h_j as $f = (\frac{\theta_j}{\frac{\pi}{2}})^2 + (\frac{h_j}{d_j \tan\beta_j})^2$. Then, we simplify the formula and transfer the layer layout problem as a nonlinear least squares optimization problem:

$$\begin{aligned} \min_{\theta_j} \quad & f(\theta_j) = \omega_\theta \left(\frac{2\theta_j}{\pi}\right)^2 + \omega_h \left(\frac{\cot\theta_j}{\tan\beta_j + \cot\theta_j}\right)^2 \\ \text{s.t.} \quad & 0 < h_j(\theta_j) < d_j \tan\beta_j \\ & 0 < \theta_j < \frac{\pi}{2} \end{aligned}, \tag{4}$$

where ω_θ and ω_h are two weighted parameters. In our implementation, both parameters are regarded as one, indicating that the *compactness* and *alignment* have the same priority. We solve this nonlinear optimization to obtain the best location of $layer_j$. In our algorithm, we need to optimize every θ to obtain the optimal position of each layer. We repeat this process from the bottommost to the topmost layer to ensure correctness.

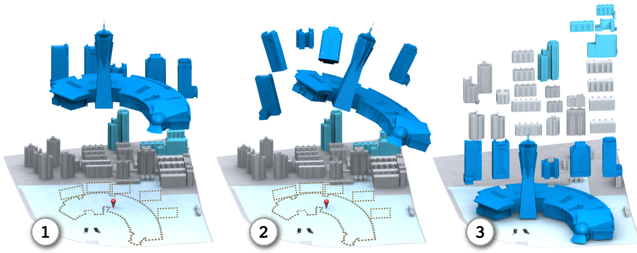

Figure 3: Results of different generation methods for exploded views: (1) result generated by our method, wherein the POI is shown without occlusion and the buildings are organized clearly; (2) result generated by the method based on force-directed layout [3], wherein the buildings are scattered over the air; and (3) result generated by the method based on explosion graph [12], wherein the POI is still occluded and all buildings are exploded.

3.5 Time Performance

The performance is discussed in this section. Our method can generate exploded views in real-time without precomputation. In the first step of our method, the most time consuming part is counting the occluded data point and blocking components of each component. In the worst case, these calculations can be solved in polynomial time if all the components are stored in a space-partitioning data structure, such as a k-d tree. The second step of our method is a pluggable part whose time consuming is determined by the specific plug-in. In the last step, approximate solutions of each θ that are accurate to one decimal place are acceptable. Therefore, because the solution space is very small (900 solutions in $[0, \frac{\pi}{2}]$), brute force algorithm is used to solve this nonlinear least-squares optimization problem. To sum up, an interactive performance can be achieved to generate an exploded view in HoloLens.

4 Evaluation

Compare with other methods In the first experiment, we compare our algorithm with two famous exploded view generation algorithms, namely, an algorithm based on force-directed layout [3] (AFL) and an algorithm based on explosion graph [12] (AEG), to demonstrate its novelty. We reimplement the main ideas of AFL and AEG and apply them to the same 3D city models. The 3D city models are an area with a POI that is occluded by a national monument. We assume the users want to explode this area to observe the POI and define the models that will be exploded into two layers. Figure 3 displays the results. The first exploded view (Fig. 3 (1)) is generated by our method, wherein the POI is shown without occlusion and the buildings are organized clearly. Figure 3 (2) is generated by AFL. The focus part, the POI in the ground, is revealed without occlusion. However, the buildings are scattered haphazardly over the air. The last exploded view (Fig. 3) is generated by AEG. The POI is occluded because AEG does not aim at revealing interesting data points. Moreover, all the buildings are exploded, which confuses the users. This experiment reveals that our promising method is suitable for immersive urban analytics.

Different data and viewpoints In Fig. 4 the second experiment, we choose a central business district (CBD) area to test our generation method for exploded views with different types of interesting data. We implemented a prototype system on the HoloLens platform. As shown in Fig. 4 (1), the POIs the user wants to analyze are hidden in the dense skyscrapers. We explode the city to reveal the hidden POIs for users. Figure 4 (2) shows the result of the explosion. The bottommost layer is the ground with insignificant content, that is, the components with low energy that provide a background context of the area. The second layer is the focus part. A region (RA), embedded with dense POIs and road networks is exploded to the second layer because its energy of RA is higher than the predefined threshold. The buildings blocking RA are also exploded to the second layer because the buildings inherit energy from RA. This result demonstrates that our algorithm can retain the blocking relationships among various components. The third layer contains the buildings that occlude the POIs in RA. These buildings are exploded to the third layer because they have the highest energy, which reveals all the hidden POIs for users to analyze. All layers are automatically placed in a position that ensures visibility, compactness and alignment. From this exploded view, the user can clearly observe and analyze the interesting data, namely, the POIs, without any occlusion and remain aware of the CBD environment which consist of dense high-rise.

Our algorithm is robust and can explode components on the basis of different interesting data with minimal input. Figure 4 (3) shows the situation where the user explodes the city from the same viewpoint as in Fig. 4 (2) but with different interesting data for analysis. In this scenario, the user intends to analyze the congested roads rather than the POIs. Without modifying the parameters and input data of our algorithm, the user instructs the system to explode the CBD area based on the congested roads by voice command. Figure 4(3) reveals that the same region is exploded to the second layer. However, the buildings exploded to the third layer are different from the last case. When the buildings are exploded to the third layer, the congested roads are revealed by a red color.

Moreover, our algorithm can generate an exploded view based on the viewpoints of users in real time. In the immersive environment, the users may observe the 3D city models from different viewpoints. Figure 4 (4) illustrates the condition where the user explodes the same CBD area to analyze the POI data from a horizontal perspective. The 3D models are exploded in smoothly real time. The number of exploded buildings in the third layer is smaller because fewer buildings are occluding the POIs from this perspective than from the perspective in Fig. 4 (2). Moreover, the layers in Fig. 4 (4) are more compact and aligned than those of Fig. 4 (2) while maintaining the visibility of all layers. This experiment shows our method is robust and the effective.

5 Conclusion

In this work, we study and explore the visualization techniques for immersive urban analytics, which to our knowledge, is the first attempt at systemically characterizing the problem. A novel technique is introduced to create an exploded view. Our technique can automatically generate an exploded view for 3D city models depending on the viewpoints of users and the data to be analyzed. The effectiveness and usability of our exploded view technique are demonstrated by multiple experiments, including a comparison with conventional generation methods for exploded views.

Our method still involves certain limitations. The method requires predefined energy thresholds to divide the components into levels; the exploded views cannot handle the situations where users' perspective changes frequently. Several possible avenues for future work are available. Given the free navigation afforded by head-mounted display system, we can improve the method to generate dynamic exploded views. A predict model for the users' perspective can be integrated into our method to update the exploded view dynamically and smoothly. The generation method for exploded view can be extended to explode multiple city models simultaneously for visual comparison purposes. We also plan to use our exploded view technique in an immersive urban analytics application and evaluate its performance by using a real-world data set.

References

[1] M. Agrawala, D. Phan, J. Heiser, J. Haymaker, J. Klingner, P. Hanrahan, and B. Tversky. Designing effective step-by-step assembly instructions. *ACM ToG*, 2003.

[2] B. Bach, R. Dachselt, S. Carpendale, T. Dwyer, C. Collins, and B. Lee. Immersive analytics: Exploring future interaction and visualization technologies for data analytics. In *ACM Proceedings on Interactive Surfaces and Spaces*, 2016.

[3] S. Bruckner and M. E. Gröller. Exploded views for volume data. *IEEE TVCG*, 2006.

[4] M. Cordeil, T. Dwyer, and C. Hurter. Immersive solutions for future air traffic control and management. In *ACM Proceedings on Interactive Surfaces and Spaces*, 2016.

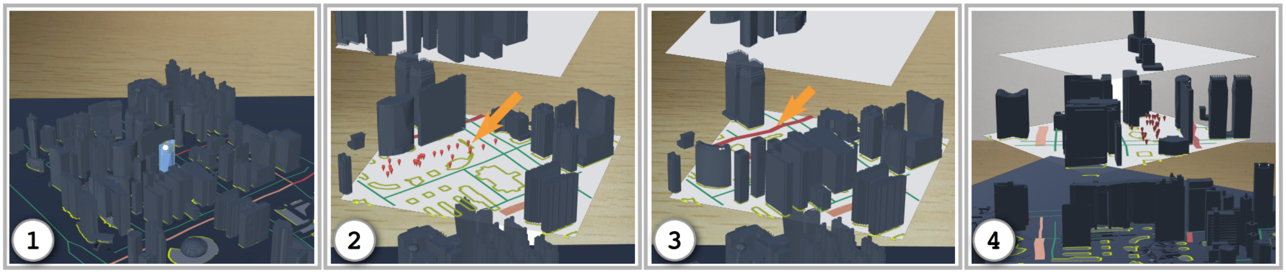

Figure 4: Exploded views in various conditions. (1) In a CBD area, the interesting data are hidden in the dense skyscrapers. (2) Some buildings are exploded to reveal the hidden POIs (the red points pointed by an arrow). (3) From the same perspective, different buildings are exploded when the analyzed target are changed from POIs to congested roads (the red road pointed by an arrow). (4)From a different perspectives, fewer buildings are exploded.

[5] ESRI. Arcgis. `http://www.esri.com/arcgis/`, 2017.
[6] N. Ferreira, M. Lage, H. Doraiswamy, H. T. Vo, L. Wilson, H. Werner, M. Park, and C. T. Silva. Urbane: A 3D framework to support data driven decision making in urban development. In *IEEE Conference on Visual Analytics Science and Technology*, 2015.
[7] Google. Earth. https://www.google.com/earth/, 2017.
[8] H. Guo, Z. Wang, B. Yu, H. Zhao, and X. Yuan. Tripvista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection. In *IEEE Pacific Visualization Symposium*, 2011.
[9] X. Huang, Y. Zhao, C. Ma, J. Yang, X. Ye, and C. Zhang. Trajgraph: A graph-based visual analytics approach to studying urban network centralities using taxi trajectory data. *IEEE TVCG*, 2016.
[10] O. A. Karpenko, W. Li, N. J. Mitra, and M. Agrawala. Exploded view diagrams of mathematical surfaces. *IEEE TVCG*, 2010.
[11] O. Kwon, C. Muelder, K. Lee, and K. Ma. A study of layout, rendering, and interaction methods for immersive graph visualization. *IEEE TVCG*, 2016.
[12] W. Li, M. Agrawala, B. Curless, and D. Salesin. Automated generation of interactive 3d exploded view diagrams. *ACM ToG*, 2008.
[13] D. Liu, D. Weng, Y. Li, J. Bao, Y. Zheng, H. Qu, and Y. Wu. Smartadp: Visual analytics of large-scale taxi trajectories for selecting billboard locations. *IEEE TVCG*, 2017.
[14] A. Moran, V. Gadepally, M. Hubbell, and J. Kepner. Improving big data visual analytics with interactive virtual reality. In *IEEE High Performance Extreme Computing Conference*, 2015.
[15] NASA. World wind. `https://worldwind.arc.nasa.gov/`, 2017.
[16] T. Ortner, J. Sorger, H. Steinlechner, G. Hesina, H. Piringer, and E. Gröller. Vis-A-Ware: Integrating Spatial and Non-Spatial Visualization for Visibility-Aware Urban Planning. *IEEE TVCG*, 2017.
[17] H. Qu, H. Wang, W. Cui, Y. Wu, and M. Chan. Focus+context route zooming and information overlay in 3d urban environments. *IEEE TVCG*, 2009.
[18] R. Sadana, V. Setlur, and J. Stasko. Redefining a contribution for immersive visualization research. In *ACM Proceedings on Interactive Surfaces and Spaces*, 2016.
[19] G. Sun, R. Liang, H. Qu, and Y. Wu. Embedding spatio-temporal information into maps by route-zooming. *IEEE TVCG*, 2016.
[20] Taqtile. Holomaps. `http://www.taqtile.com/holomaps/`, 2017.
[21] M. Tatzgern, D. Kalkofen, and D. Schmalstieg. Multi-perspective compact explosion diagrams. *Computers and Graphics*, 2011.
[22] X. Tong, C. Li, and H. Shen. Glyphlens: View-dependent occlusion management in the interactive glyph visualization. *IEEE TVCG*, 2017.
[23] Wikipedia. Dbscan. `https://en.wikipedia.org/w/index.php?title=DBSCAN&oldid=769169609`, 2017.
[24] Y. Zheng, W. Wu, Y. Chen, H. Qu, and L. M. Ni. Visual analytics in urban computing: An overview. *IEEE Transactions on Big Data*, 2016.