VU-Flow: A Visualization Tool for Analyzing Navigation in Virtual Environments

Luca Chittaro, Roberto Ranon and Lucio Ieronutti

Human-Computer Interaction Lab
Department of Math and Computer Science, University of Udine
via delle Scienze, 206, 33100 Udine, Italy

{chittaro, ranon, ieronutt}@dimi.uniud.it

Abstract—This paper presents a tool for the visual analysis of navigation patterns of moving entities, such as users, virtual characters or vehicles in 3D Virtual Environments (VEs). The tool, called VU-Flow, provides a set of interactive visualizations that highlight interesting navigation behaviors of single or groups of moving entities that were the VE together or separately. The visualizations help to improve the design of VEs and to study the navigation behavior of users, e.g., during controlled experiments. Besides VEs, the proposed techniques could also be applied to visualize real-world data recorded by positioning systems, allowing one to employ VU-Flow in domains such as urban planning, transportation, and emergency response.

Draft version of the paper “VU-FLOW: A VISUALIZATION TOOL FOR ANALYZING NAVIGATION IN VIRTUAL ENVIRONMENTS” published in "IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, SPECIAL ISSUE ON VISUAL ANALYTICS".

1 INTRODUCTION

The development of visualizations that support the analysis of physical or virtual spaces, their inhabitants, navigation and interaction patterns is important in a variety of domains, such as social sciences, geography, architecture, and emergency response. For example, visualizations that allow an analyst to understand how people navigate roads and buildings could play an important role in the design, evaluation and optimization of such spaces (e.g., planning transports or evacuation procedures).

In the case of virtual spaces, a number of visualizations of navigation and interaction patterns have been proposed, mostly targeted at Web sites and hypermedia spaces [1][2][3][4]. A few recent projects [5][6][7][8], however, have focused on visualizing navigation and interaction behaviors in 3D Virtual Environments (hereinafter, VEs). The case of VEs is interesting for a number of reasons. First, compared to physical environments, it is easier and cheaper to acquire detailed and accurate navigation and interaction data. At the same time, visualizations developed for VEs are likely to be effectively applicable to their real-world counterparts and the increasing volume of real-world spatio-temporal data collected through positioning systems such as GPS. Second, VEs are becoming popular in many domains. For example, online VEs are increasingly used for education, entertainment and social interaction, and some of them have reached a size that is comparable to their physical counterparts. Designing usable and navigable VEs is a difficult task, and there is a lack of tools that help one in understanding the effects of design choices.

This paper presents a visualization tool, called VU-Flow (Visualization of Users' Flow) that can be used to analyze the navigation patterns of moving entities such as users, vehicles or virtual characters in VEs. More specifically, starting from logged navigation data, VU-Flow is able to produce interactive visualizations allowing an analyst to answer questions such as:

- Do users (or groups of users) *X* and *Y* have similar navigation patterns?
- Which areas are more traveled by users?
- Which objects or points of interest get more users' attention?

- How a certain area is typically visited by users ?

While similar approaches [5][6] focus on visual analysis of group behaviors in multi-user VEs, VU-Flow can be employed for both multi-user (e.g., online communities) and single-user VEs (e.g., stand-alone games and Virtual Reality applications, as well as Web3D sites), also allowing one to compare navigation data recorded in different sessions. Since visual analysis of several moving entities is a difficult task, VU-Flow includes a set of visualizations that exploit color-coded overlays and hedgehog arrows to visualize global navigation behaviors of multiple entities.

The paper is structured as follows: section 2 motivates the need for visualization of spatio-temporal behaviors and interactions in VEs; section 3 describes related work; section 4 presents VU-Flow and its visualizations; section 5 demonstrates how the visualizations can help in discovering navigation patterns, considering an online VE as a case study; section 6 discusses current limitations of VU-Flow and introduces future work.

2 MOTIVATIONS

In general, analyzing how people navigate and interact in a VE can be interesting both for psychological studies of users and for optimizing VEs for navigation and interaction. It is indeed well-known that VEs tend to suffer from usability problems such as user disorientation, difficulties in wayfinding and movement control.

Tools for the visual analysis of users' activities in VEs can be generally useful to:

- classify users' visiting styles or preferences (e.g., one can discover that most users visit the VE following an unexpected path and thus they encounter points of interest and objects in a sequence that is very different from those intended by VE designers);
- compare different design choices (e.g., different signs or different placements of landmarks and signs to help orientation in a virtual city), or assess the navigability of a VE (e.g., how many users manage to easily reach a certain location? How many get lost?);
- evaluate the effectiveness of electronic navigation aids (e.g., 2D vs. 3D maps) or more generally evaluate interaction techniques (e.g., different ways of controlling movement).

Multi-user VEs present additional opportunities, such as studying how users interact among each other, and helping in the design of VEs that support specific types of group activities. For example, in educational multi-user VEs, researchers and educators are interested in how social groups form, interact, and vanish, or what educational purposes can be well served by using VEs as opposed to classroom activities [5]. In multi-user online combat games, instead, one may be interested in understanding competitive behaviors or team coordination strategies and tactics [6].

Navigation and interaction visualization may be also used during VE visits [5], e.g. for social activities support. For example, in educational VEs, teachers might be interested in knowing how students are going through a certain learning experience, or simply need to know if a student got lost or otherwise needs help. In this scenario, real-time visualizations of students' navigation and interaction may allow the teacher to rapidly detect problems and react accordingly.

Finally, besides users, VEs may be inhabited by simulated entities able to move and interact that need to be taken into account in the analysis. For example, VEs that simulate evacuation procedures, traffic flows or military operations typically employ (crowds of) simulated moving and interacting entities (e.g., virtual humans and vehicles). In these cases, analysts could be interested in both real-time and post-event visualizations able to highlight the intensity of entities' flow in a certain area or critical situations such as traffic congestions.

3 RELATED WORK

In the latest years, there have been several attempts at providing visualizations of Web navigation and interaction logs for various purposes, such as improving usability [1], analyzing e-commerce clickstream data [2], and highlighting interesting patterns [3][4]. However, these approaches are not concerned with VEs: they focus on 2D Web sites and hypermedia, and are therefore mostly based on algorithms for visualizing abstract graph structures [1].

A few projects have instead focused on visualizing navigation and interaction behaviors in VEs. The proposed visualizations are typically based on superimposing icons and glyphs or color-coded overlays over a 2D map of the VE. A good account of related research areas (such as social and scientific visualization) is provided in [5]. In the following, we illustrate in detail those approaches that specifically target VEs.

Börner et al. [5] focus on visualizing the evolution of virtual communities in multi-user online VEs for education built using the *ActiveWorlds* platform [9]. They record spatially and temporally referenced user interactions

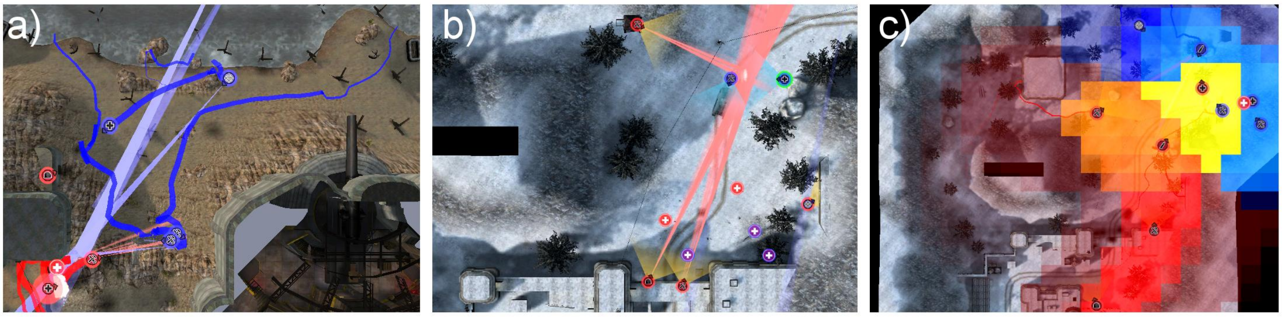

Fig. 1. (a) visualization of players' class, paths, and health [6]; (b) visualization of fire traces and fields of view [6]; (c) occupancy map for a game with a blue and a red team (yellow and orange zones highlight active conflicts) [6].

such as navigation (including teleporting), object manipulation (e.g. creation, modification and deletion of virtual buildings), Web access (i.e. activation of Web hyperlinks available inside the VE) and chatting [5]. From this data, they build a 2D map of the VE that highlights buildings, hyperlinks, and teleport locations. Then, navigation and chat logs are visualized over the map. In particular, navigation data is visualized by plotting the trails of each user as polylines whose color can encode either the temporal sequence of user positions or different users' trails. A problem of this visualization, as acknowledged by its authors, is that it becomes easily cluttered and unreadable as the number of users increases. The two solutions proposed are:

- computing *diffusion potentials*, i.e., divide the map into a grid of rectangular cells, compute the average of all 2D vectors that describe user movements for each cell, and visualize it over the map as a small arrow;
- clustering users into social groups, and plot the aggregated trail of an entire group using a single polyline, where the thickness of each part of the polyline depends on the average proximity of users.

Chat data are visualized in 3D as data hills (whose height indicates the number of chats) over the 2D map of the environment.

From an implementation point of view, the 2D map is computed starting from the so-called ActiveWorlds *Registry* file, listing all objects (and their positions) in a specific ActiveWorlds VE, while visualizations of user activities are computed from ActiveWorlds *bot log* files, logging users' movements, actions, and chats.

Hoobler et al. [6] focus on team-based combat games in multi-user online VEs and on the visual analysis of team strategies and competitive behaviors. Their *Lithium* system is able to acquire interaction data, such as players' movements, fire actions and status (e.g., current health), and visualize them over a bird's eye orthographic view of the VE as the match progresses. The visualizations are divided into two categories:

- *local visualizations* highlight players' positions (using icons to encode players' classes, e.g., medics, and status, e.g., health, as in Fig. 1a), paths (using polylines whose thickness is proportional to temporal recency, as in Fig. 1a), fire traces (using triangles that fan out from the firing player, as in Fig. 1b), and fields of view (by drawing cones of vision);
- *global visualizations* use color-coded overlays, made by a grid of rectangular cells, where the color of each cell is computed from game data. For example, the *occupancy map* (see Fig. 1c) highlights which team has more recently occupied each cell (the color of the cell indicates the team or a zone of active conflict, while brightness indicates how recently the team was there). Other visualizations are more game-specific, e.g. highlighting areas from which support fire is coming or where medics are positioned with respect to casualties.

The *Lithium* system is a client for the online game *Enemy Territory*. By using an application-specific API, it visualizes the game using an overhead orthographic view, and draws on it screen-aligned icons (for local visualizations) and 2D overlays (for global visualizations) that are updated at every frame as the game progresses.

4 The VU-Flow Tool

The approaches presented in the previous section are targeted at specific kinds of online multi-user VEs. From a visualization point of view, their strongest feature is the analysis of group behaviors (e.g., group formation and evolution, chat, team strategies), and the handling of application-specific events, such as adding a building or firing a gun. From an implementation point of view, the possibility of visualizing such events comes at the price of flexibility, since it is not straightforward to employ these approaches with data recorded in VEs built for different purposes or with other technologies.

By contrast, VU-Flow has been designed as a general-purpose tool for *off-line* (i.e., post-visit) visual analysis of *movements and navigation* in any VE application. The tool post-processes logs of movements, i.e. follows an approach commonly used in areas such as security, performance, and network analysis.

VU-Flow shares with the approaches mentioned in the previous section the general idea of visualizing users' behavior over a 2D map of the VE, and therefore its visualizations are accurate when the user stands on the VE terrain (e.g., walking, running, driving a car,…), and less effective for free movement in 3D such as flying. Moreover, instead of targeting application-specific events, its main focus is on providing interactive visualizations that help one in answering general navigation-related questions.

In this section, we first describe the data that VU-Flow handles and the employed 2D map of the VE; then, we present in detail the available visualizations and discuss their implementation. Visualization examples will be taken from 3 real studies we carried out using VU-Flow: a study of visitors' behavior in a single-user 3D virtual museum of computer science that presents computing equipment from the seventies (Fig. 2 and Fig. 4); a comparative analysis of the effects of different electronic navigation aids on the navigation performance of users who have to reach a target in a single-user VE (Fig. 3 refers to users' behavior with one of the aids); a congestion study of a multi-user VE representing a real-world building (Fig. 5).

4.1 Navigation logs

VU-Flow derives its visualizations from log files (hereinafter, *navigation logs*) in which every line describes the position of a moving entity at a certain time. More specifically, each line of a navigation log is structured as follows:

- identifier of the moving entity (e.g., user name);
- recorded position and orientation in 3D space, with respect to a global coordinate system;
- timestamp of the recorded position;
- optionally, an identifier of a navigation session; this can be used by the system to distinguish navigation data belonging to the same user, but referring to different visits to the VE.

Most open VE technologies allow one to record navigation logs as those required by VU-Flow; for example, we can easily obtain navigation logs from any VRML [10] and X3D [11] VE as described in [8]. However, with proprietary VE technologies, and especially on-line games, obtaining navigation logs could not be as straightforward [6].

4.2 Map of the VE

VU-Flow employs a 2D map of the VE both as the graphical background over which the visualizations are drawn, and as a data structure used in the computation of most visualizations. For example, in the visualization of Fig. 4b, VU-Flow exploits knowledge about the structure and the position of objects in a room map to produce a visualization that highlights what has been more seen by users.

The analyst has to provide VU-Flow with:

- a text file that describes a *n* x *m* matrix of binary values. Each cell of the matrix corresponds to a square area of the VE, and the value of the cell indicates whether the area is navigable (i.e., it does not contain obstacles such as walls) or not. VU-Flow uses this matrix also to derive a simple black and white picture of the map (as the ones we use throughout the paper) to be used as a visualization background.
- optionally, a bird's eye picture of the VE to replace the default black and white map as visualization background. This picture can be obtained by taking a screenshot of the VE from a bird's eye position or by orthographic projection into a texture [12] or by drawing it with a graphics program.
- a reference point in the map and its location in the global coordinate system to which navigation logs refer, used to correctly transform navigation data into correct positions on the map.

In many cases, the required *n x m* matrix can be automatically obtained, e.g. by using image processing algorithms on a map picture [12]. In X3D/VRML VEs, one can use a map derivation technique [13] that directly calculates the matrix from the VE.

In the case of multi-floor VEs, such as some buildings, VU-Flow deals with the different floors individually: for each floor, it needs the above described data, and each line of the navigation log must indicate the floor it refers to.

4.3 Visualizations

Once the 2D map of the VE and the navigation logs have been loaded, VU-Flow is able to provide two categories of visualizations:

- *non-aggregated visualizations,* aimed at highlighting navigation patterns of individual moving entities, e.g.,

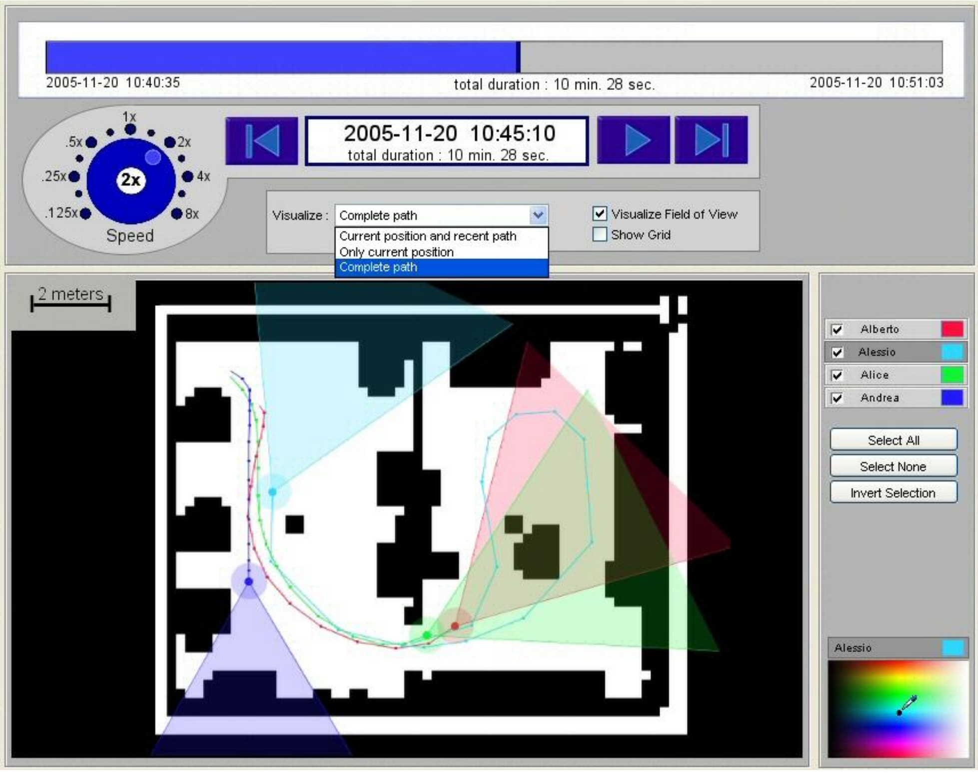

Fig. 2. Visualization of individual paths and current fields of view using VU-Flow.

to compare navigation patterns of users *X* and *Y*; data belonging to different moving entities are highlighted separately over the 2D map;
- *aggregated visualizations*, aimed at highlighting a population's navigation patterns, e.g., identifying more traveled areas in the VE. In these visualizations, data belonging to different moving entities are first aggregated, and then visualized over the 2D map.

Non-aggregated visualizations

Non-aggregated visualizations plot the trail of each moving entity using points on the map to visualize logged positions, lines to connect them, and triangles to highlight the field of view of the entities (as in Fig. 2). Colors of points and lines are automatically chosen so that each moving entity is associated to a different color. The analyst can:

- moving the considered time instant, by using a slider in the timeline (top part of Fig. 2) that starts (ends) with the less (most) recent timestamp in the considered navigation logs;
- replay movements using the VCR-like controls shown in Fig. 2, that include fine tuning of replay speed;
- choose which entities are visualized (using the checkboxes in the right part of Fig. 2), how paths are drawn (positions only, full paths, recent parts of paths), and whether to show the fields of view of the entities;
- zoom and pan to analyze specific parts of the visualization in more detail.

Non-aggregated visualizations can also work in a *relative-time* mode, that is useful when one wants to analyze navigation logs that were recorded at different times. For example, one may need to compare different sessions concerning the same user, or different users that navigated the VE in single-user mode. In the relative-time mode, all the chosen sessions are visualized as if they started at the same instant, e.g., allowing the analyst to easily highlight what users do just after entering the VE. Section 5 gives practical examples of how these visualizations are used.

For X3D/VRML VEs, VU-Flow also offers the possibility of visualizing paths inside the VE itself, instead of the 2D map, using the controls we described above. This feature has been described in [7].

Aggregated visualizations

Drawing detailed paths on the map helps one in analyzing single entities, but the visualization becomes cluttered and confusing when considering a user's very long and complex path or a population of several users (e.g., Fig. 3a and Fig. 9a). In these cases, there is the need for visualizations that present navigation data at a level of abstraction that makes it easy to visually detect peculiar behaviors or patterns. For this reason, VU-Flow provides the analyst with five types of aggregated visualizations, respectively highlighting:

- *time spent*: areas in which entities spent more/less time, e.g., Fig. 3b;
- *traveled areas:* more/less traveled areas, e.g., Fig. 3c; an area is more traveled as more entities pass through it or the same entity passes through it multiple times;
- *seen points:* more/less seen objects and locations, e.g., Fig. 4b;
- *congestions*: intensity of traffic congestions, e.g., Fig. 5b;
- *flow*: flow of moving entities, e.g., Fig. 3d, 3e, 3f.

The visualizations are obtained by inserting colored hedgehog arrows into the n x m cells of the map (for the *flow* visualization) or filling cells with colors (for the other visualizations). The analyst can customize color scales and size of the hedgehog arrows, as well as zoom and pan to better focus on specific parts.

We now discuss in detail the properties and usefulness of aggregated visualizations.

The *time spent* visualization uses a color scale to indicate how much time was spent by entities in each area of the VE. As a result, it takes into account speed of entities, because the slower they move, the longer is the time to pass through an area. For example, Fig. 3b highlights in brighter red an area on the left where users spent time to choose among 3 different ways.

The *traveled areas* visualization uses a color scale to visualize how many times entities passed through each area of the VE. Unlike the *time spent* visualization, that is affected by how much time an entity spent in any given position (and is thus *time-dependent*), the *traveled areas* visualization is *time-independent*. As a result, it is not affected by the speed of the entities. For example, the fact that Fig. 3c highlights in brighter red an area on the left similar to Fig. 3b means that users went back and forth to look at the 3 possible ways to go from that area. As a second example, note that there is a red area in Fig. 3b that starts from the bright red area on the left and follows the correct path to the target, while the same area is bluish in Fig. 3c. This is due to the fact that most users followed the right path to the target without coming back, but proceeded slowly after choosing among the 3 ways and later accelerated. The slower speed is probably related to VE structure: the considered red area corresponds to a narrow corridor with turns.

The *seen points* visualization uses a color scale to visualize how much time was spent by entities looking towards objects (i.e., non-navigable areas in the VE map). Since objects may only partially fall into an entity's field of view or they may fall into it although they are very far, this visualization takes also into account the distance of objects from the entity and how central is the position of the object in the field of view to rate how much an object has been seen. Finally, since the map provides positions of objects, VU-Flow considers possible occlusions of sight; however, since the map is bi-dimensional, this might not always give precise results, since occlusions might be overestimated. Fig. 4 shows an example of *seen points* visualization for the 3D virtual museum application. The visualization allows one to easily determine which exhibits have been more/less seen, e.g. among the exhibits lined up to the wall in the upper part of the figure, the printer (i.e., the rightmost exhibit) was the most seen.

The *congestions* visualization is useful when multiple entities are navigating the VE simultaneously. It highlights areas of the VE where several entities traveled at the same time, and thus were likely to get in each other's way (e.g., trying to pass through narrow passages, such as doors or corridors, or to examine the same object at the same time). The severity of a traffic congestion depends on the density of entities in a given area, and on the amount of time during which that density is high (this is thus a time-dependent visualization). Fig. 5b shows an example that considers the paths shown in Fig. 5a: a severe traffic congestion is highlighted near the upper passage. Note that by looking only at the paths in Fig. 5a, one could have hypothesized congestion problems also for the left passage, but Fig. 5b allows to rule that out.

While the above described visualizations do not show the direction of movement, *flow* visualizations inform about it using colored hedgehog arrows (e.g., Fig. 3d and Fig. 3f) and streamlines (e.g., Fig. 3e). More specifically, the direction of each arrow is the sum of all directions of movement from the considered area, while the color of the arrow is associated either to the intensity of flow in the given direction (e.g., Fig. 3d) or to the standard deviation of all considered movement directions (e.g., Fig. 3f). In the first case, the visualization highlights most followed directions of movement in each area of the VE. For example, Fig. 3d shows that users generally managed to follow the optimal path to the target. The standard deviation of the direction highlights instead how closely entities chose the direction indicated by the arrow. For example, Fig. 3f is able to effectively highlight: (i) preferred directions in areas traveled by only a few users; (ii) areas where users were hesitant, or had to step back because they realized to be in the wrong direction. These flow visualizations do not take into account the speed of entities, and are thus time-independent.

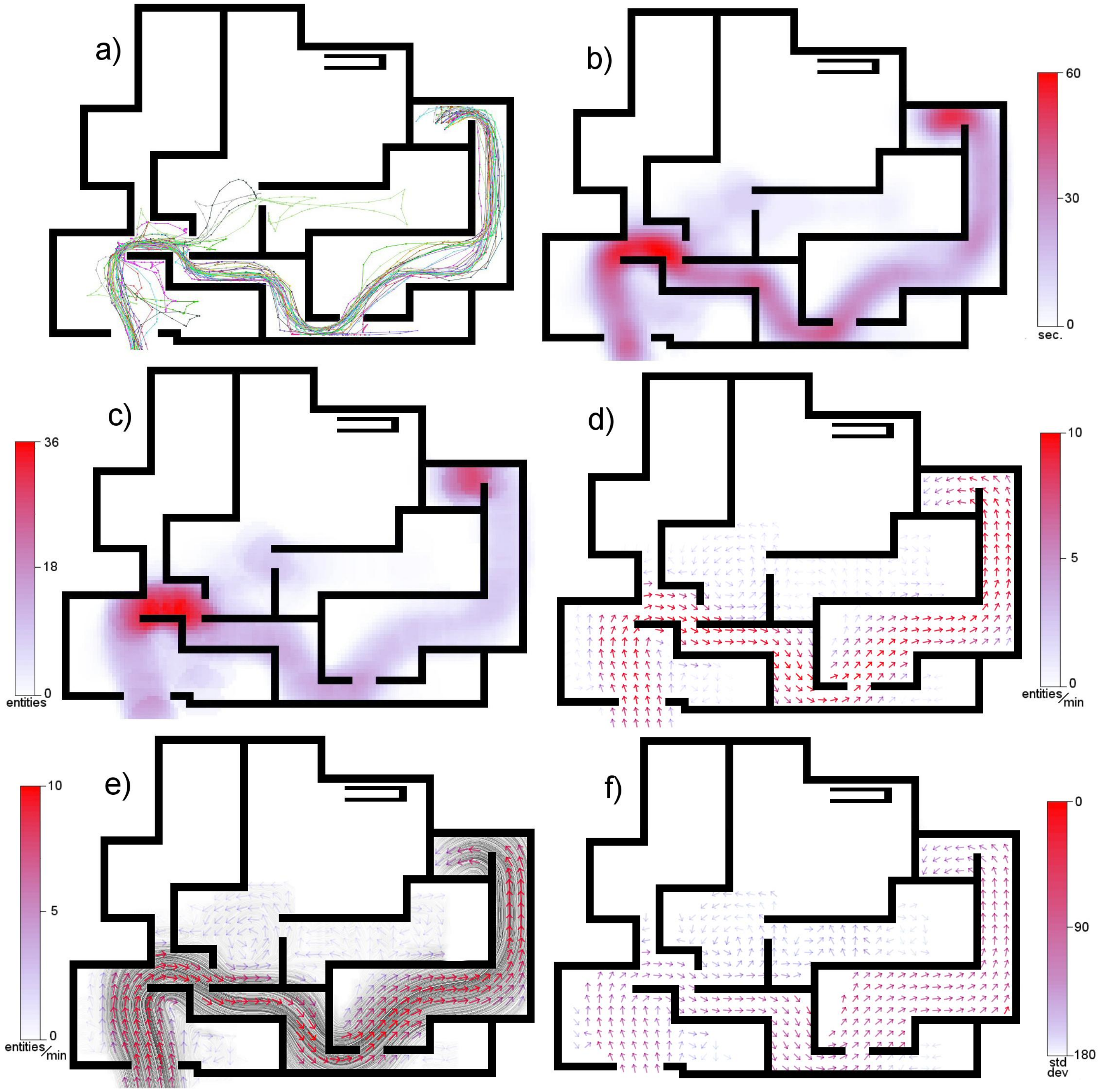

Fig. 3. Different VU-Flow visualizations of the same dataset, concerning an experiment involving 30 users that had to reach a target in a virtual building with the help of an electronic navigation aid (users start at the bottom left, and the target is in the top right of the map): (a) full users' paths; (b) time spent; (c) traveled areas; (d) users' flow (color indicates flow intensity); (e) users' flow (color indicates flow intensity) using the *Enhanced IDraw* technique [18]; (f) users' flow (color indicates standard deviation of movement directions).

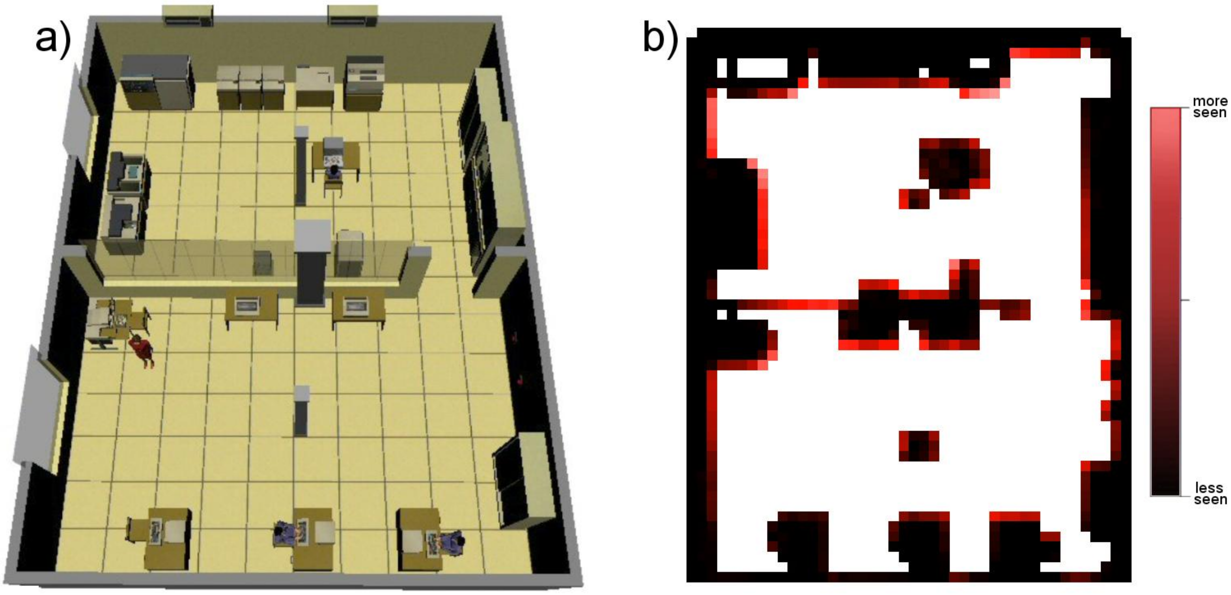

Fig. 4. (a) 3D virtual museum of computer science [14]; (b) *seen points* visualization derived from 28 visitors' logs.

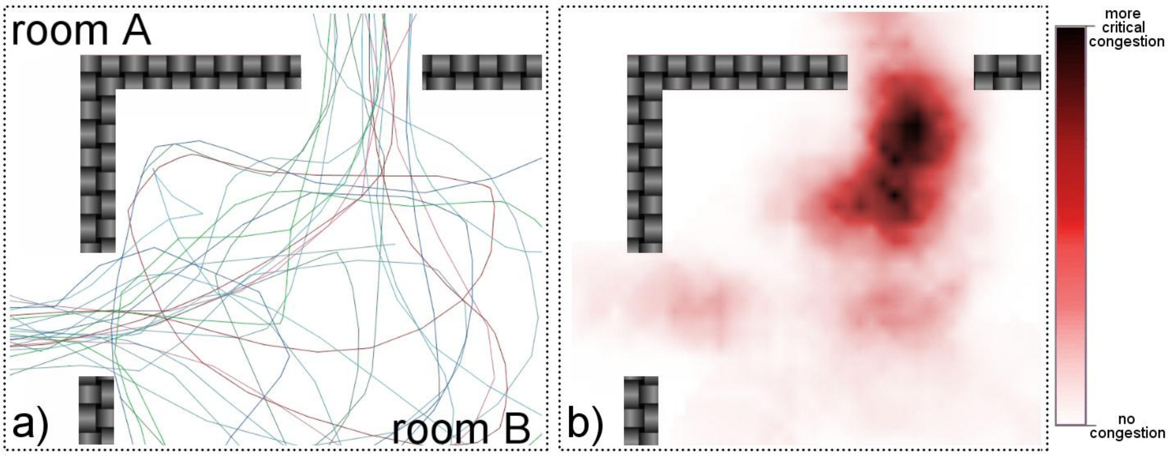

Fig. 5. (a) A set of users' paths in a part of a virtual building; (b) *congestions* visualization derived from the same data.

4.4 Visualizing Spatio-Temporal Subsets of Navigation Data

In some cases, the analyst needs to concentrate on particular temporal intervals and/or specific areas of the VE. For example, in an online VE, she might want to know what happened during the last Christmas season, or concentrate on a recently added or a scarcely visited area.

VU-Flow allows one to select a subset of navigation data by means of spatio-temporal queries. More specifically, the analyst can specify a set of one or more temporal intervals through temporal sliders (upper part of Fig. 6), while rectangular areas of interest can be selected by dragging the mouse in the appropriate areas of the 2D map (lower part of Fig. 6). In the case shown in Fig. 6, the analyst has focused on the days that precede Christmas in the last two years, and on two (overlapping) parts of the VE.

4.5 Implementation

VU-Flow is mainly composed by three modules, called *Filtering, Analysis* and *Visualization* (see Fig. 7). The Filtering module estimates the position of entities in time between each pair of consecutive logged positions, using information about:

- maximum moving speed of entities (which can be either manually specified or estimated from an overall

analysis of navigation logs);
- areas of the VE which are not navigable, and thus where an entity cannot be (this information is contained in the previously described *n* x *m* matrix).

For example, the Filtering module can identify and mark teleport movements (typically characterized by two positions that are far in space but very close in time) which could lead to erroneous visualizations. The Filtering module is also used to filter out data that do not match spatio-temporal queries (see previous section).

The Analysis module processes filtered data to set proper data structures for the visualizations. Visualizations that use only color scales are based on a discrete scalar field, while flow visualizations are based on a discrete vector field. For example, the *time spent* visualization requires the computation of a *n* x *m* scalar field, where each scalar value is the time in seconds during which entities were in the corresponding area. Algorithms used by the Analysis module are strongly inspired by *Time Geography* concepts [15][16]. More specifically, we use the concepts of *space-time path* as the basis for non-aggregated visualizations and *space-time prism* as the basis for aggregated visualizations. In particular, considering an entity traveling on a surface with maximum speed v_{max}, starting from an initial position p_i at time t_i, and arriving at final position p_j at time t_j, the space-time prism allows one to define an elliptical region of space containing all positions the entity might have been during the temporal interval [t_i, t_j]. The elliptical region is mapped into cells of our *n* x *m* matrix and used for computing the needed visualization. This allows us, compared to just using points in the segment with endpoints p_i and p_j (i.e., linear interpolation), to improve accuracy in the computation of the scalar and vector fields used for visualization.

An additional functionality of the Analysis module is to produce a vector of quantitative data (called *feature vector*) characterizing navigation behavior of an entity. The feature vector contains:
- average and standard deviation of moving speed;
- average and standard deviation of angular speed;
- number of self-intersections in the navigation paths.

Feature vectors can be exported for later processing by other data analysis applications or machine-learning techniques.

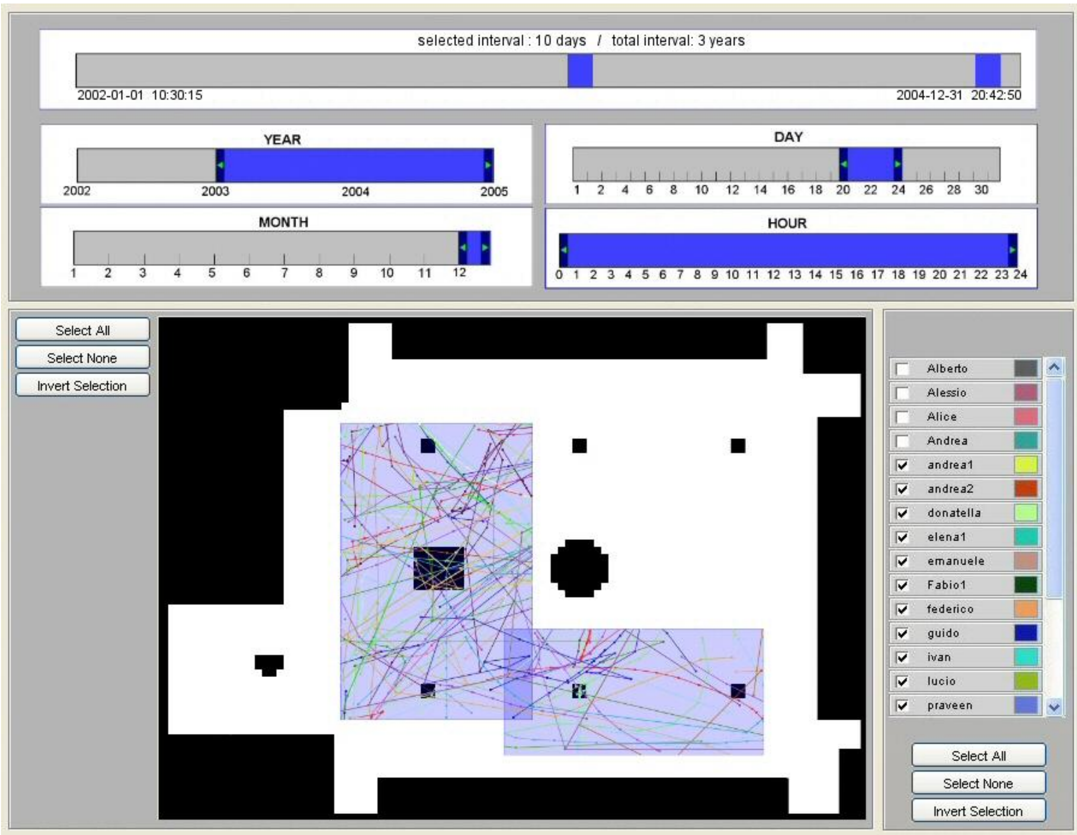

Fig. 6. Interface for spatio-temporal queries on navigation data in the VU-Flow tool.

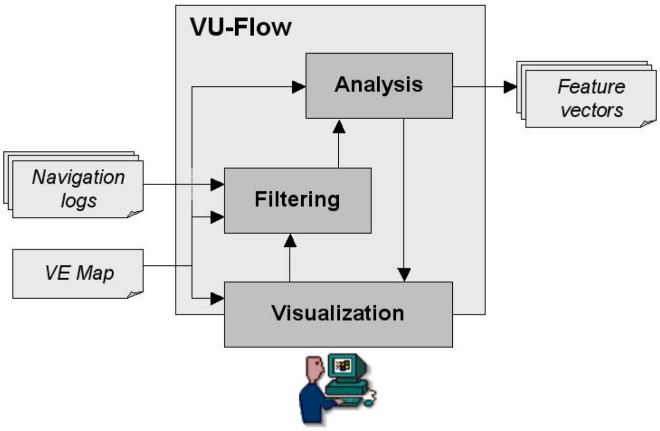

Fig. 7. Architecture of VU-Flow.

Fig. 8. A screenshot of Udine3D [20] from the user initial position and orientation.

The Visualization module is based on standard techniques for visualizing scalar and vector fields. In particular, we use hedgehog arrows [17] for vector fields, and the *Enhanced Integrate and Draw* (*IDraw*) technique [18] for streamlines. IDraw filters a white noise texture along the path of streamlines to derive a dense visualization of a discrete vector field.

Moreover, we use color-coding mechanisms [19] that allow the analyst to:

- choose between using two or three control points (each one associating a value with a color), and select appropriate control points;
- fine-tune color coding, by controlling the transitions between colors. In particular, the analyst can set the parameter used for exponential interpolation between colors. With respect to linear interpolation, this solution allows one to more clearly highlight subtle differences between values [19] and is particularly effective when values are not uniformly distributed (as it often happens in our case).

5 Using VU-Flow in Practice

This section presents a detailed case study where VU-Flow has been used to analyze navigation logs of *Udine3D* [20], an online VE developed in VRML. *Udine3D* (Fig. 8) allows users to take a virtual walk into one of the squares of the city of Udine, Italy, and get information on its history and major buildings. The VE does not allow visitors to see each other (i.e., it can be only visited in a single-user mode).

We collected navigation data on 130 unique visits containing a total of about 23'000 sampled data. Some of the obtained visualizations are shown in Fig. 9. The colored bar at the top of the Figure gives indications about the length of users' visits (using the relative-time mode discussed in section 4.3): for any length, color indicates how many visits lasted as long. The longest visit lasted about 11 minutes, while most visits lasted less than about 3 minutes. After loading navigation data into VU-Flow, we analyzed the automatically derived visualizations to highlight navigation problems or patterns of interest.

Fig. 9a shows full users' paths: lines going out from the map are due to the fact that a few users activated a function of their VRML plug-in that teleports to a distant location in 3D space from which the entire VE can fit into the browser window. After noticing those lines, we replayed the logs that include them with non-aggregated visualizations and noticed that the involved users tended to have problems going back to the virtual square and to get lost ("fit" options of VRML plug-ins should thus be disabled in the Udine3D VE). The non-aggregated visualization of does not offer more insight into users' navigation patterns in the Udine3D case.

In the *time spent* visualization (Fig. 9b), the bright red area in the top right corresponds to the VE entry point, where supposedly most users spent some time to look around (or learning how to navigate) before taking a precise path. Time is spent quite uniformly in the remaining parts of the VE, with some exceptions. One involves locations where visitors can get cultural information about the square (by clicking on human figures with a book in their hands): the visualization gives a quick idea of how much users stopped at these different locations. The second exception involves two areas (in the upper left and lower left of the figure) located at roads that seem to lead out from the square but actually prevent users to fall out from the VE through transparent walls associated to an explanatory message that appears as users approach. A detailed replay of logs in those areas revealed that some users repeatedly tried to go through the transparent walls.

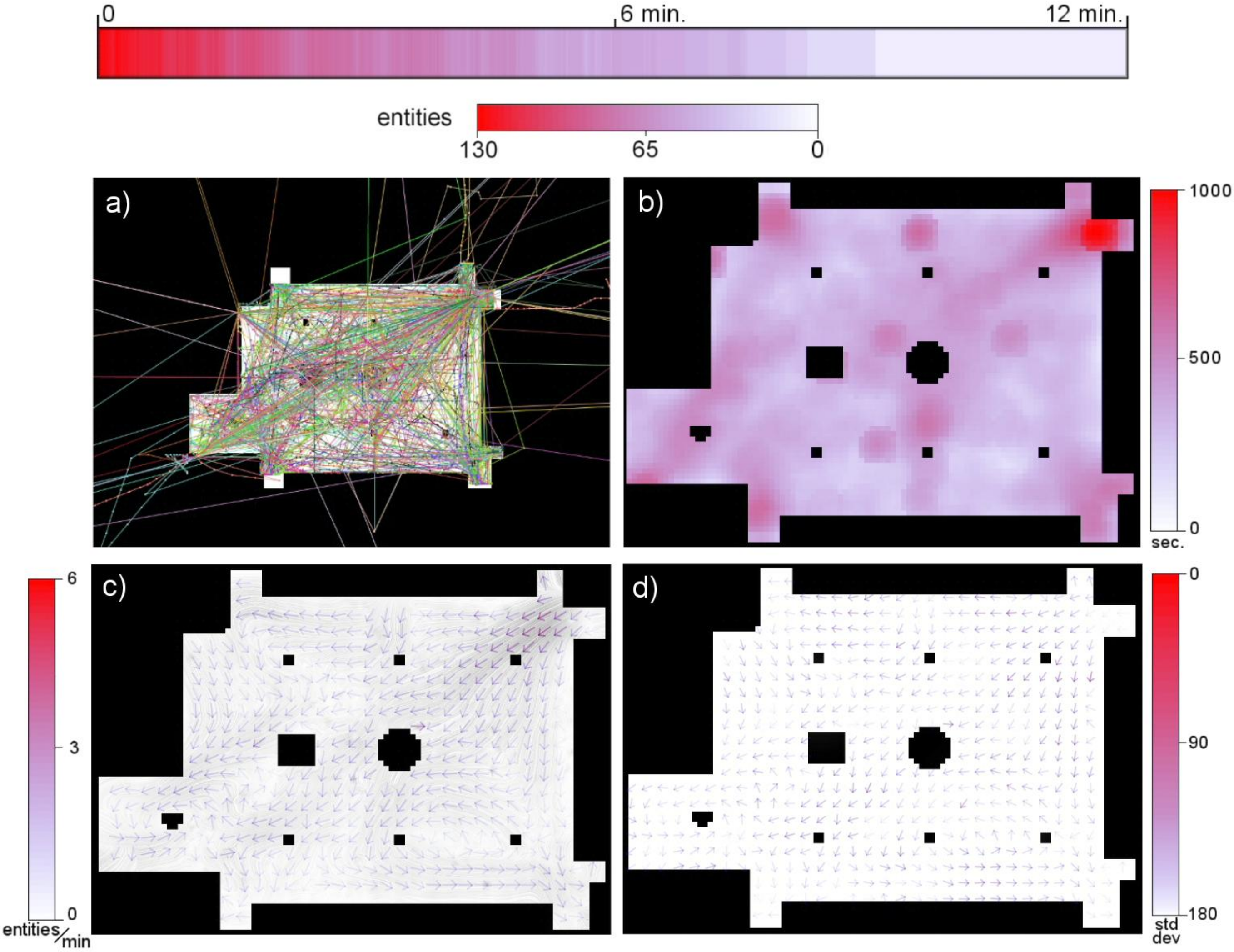

Fig. 9. VU-Flow visualizations of data collected for the Udine3D online VE [20]: (a) full users' paths; (b) time spent in the different areas; (c) users' flow (color indicates flow intensity); (d) users' flow (color indicates standard deviation of movement directions).

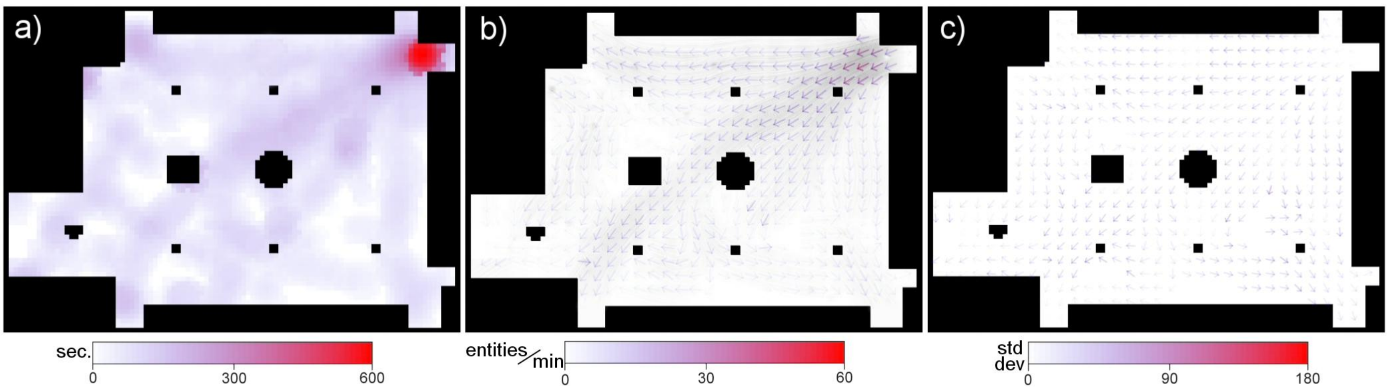

Fig. 10. VU-Flow visualizations of data collected for the Udine3D online VE [20], restricted to the first 40 seconds of each visit: (a) time spent in the different areas; (b) users' flow (color indicates flow intensity); (c) users' flow (color indicates standard deviation of flow direction).

This might be due to the fact that they did not understand the language in which the message is written or they were simply trying to check if there was a way to go through anyway.
Fig. 9c and Fig. 9d illustrate users' flow, and highlight a major navigation pattern, i.e., users seem to move from the initial position towards the fountain in the center of the square. To confirm this hypothesis, we selected the first 40 seconds of every visit through a temporal query. The resulting visualizations are depicted in Fig. 10. More specifically, Fig. 10a confirms that most users initially stood still or close to the entry point, while Fig. 10b and 10c allow one to refine the hypothesis made from Fig. 9c and Fig. 9d: while it is true that the initial preferred direction goes towards the fountain, a considerable percentage of users chose to walk along the upper edge of the VE.

6 Conclusions

Although we have only recently started using VU-Flow on significant real cases, results are very promising. The tool proved to be effective in discovering navigation patterns, highlighting critical situations, and prompting usability improvements. However, a more thorough and formal evaluation of the tool is needed. We thus plan to formally evaluate the effectiveness of the employed visualizations, as well as to apply VU-Flow to other challenging scenarios. In particular, we are experimenting with visualizations of real-world data recorded with GPS units. Unfortunately, unlike navigation data logged in a VE, position data acquired through GPS is often inaccurate, especially in some urban environments, and orientation cannot always be determined. These limitations inevitably affect the accuracy of visualizations, and needs to be counterbalanced with proper filtering and correction algorithms.

Employing VU-Flow in practical projects has also highlighted some limitations, which we discuss in the following, together with future work aimed at overcoming them.

First, some visualizations are quite costly to compute, and for realistic situations (as the case study presented in the previous section), one cannot get an immediate update of the visualizations when she changes the selected subset of data (e.g., to consider only some users). Although there is a lot of room for optimization of the current implementation (and this is one of the activities we are carrying out), it is to be seen how effectively VU-Flow will be able to handle huge amounts of data such as those produced by a massive multiplayer online VE. However, the computational complexity heavily depends on the required accuracy of visualization, and the current implementation does not allow the analyst to easily experiment with different levels of detail. Therefore, one of our goals is to study solutions for analyzing navigation data at different levels of detail (e.g., using coarser data visualizations first, and then allowing one to refine them for particular VE areas or time intervals).

In its current version, VU-Flow does not handle navigation logs as they are being produced. However, there are many scenarios, either virtual or physical (e.g., emergency management) where that capability can be very useful (e.g., for monitoring purposes). We plan to extend the tool in this direction. One option we are exploring is to modify the Analysis module to work as a process that monitors the presence of new navigation data, and incrementally updates all visualizations in the background. In this way, the analyst could see the visualizations updated in real-time, and move from one visualization to another with no significant time delays.

In many situations, navigation data becomes more meaningful if correlated with other interaction data. For example, in Udine3D visitors can click on human figures to get cultural information. Actually, VU-Flow highlights that users stop near the human figures (presumably to read the information), but it would be interesting to visualize also when and where clicks occur. Future work will be thus aimed at extending and generalizing our approach to handle and visualize other kinds of interaction data.

Acknowledgment

Our research has been partially supported by the Italian Ministry of Education, University and Research (MIUR) under the PRIN 2005 projects “Adaptive, Context-Aware, Multimedia Guides on Mobile Devices” and "VIA - Virtual Interactive Architecture".

References

[1] E. H. Chi, “Improving Web Usability Through Visualization”, *IEEE Internet Computing*, issue 2, pp. 64-71, 2002.

[2] J. Brainerd and & B. Becker, ”Case Study: E-Commerce Clickstream Visualization”, in *Proc. of the IEEE Symposium on Information Visualization 2001,* IEEE Press, 2001, pp. 153-158.

[3] I. Cadez, D. Heckerman, C. Meek, P. Smyth, S. White, “Visualization of navigation patterns on a Web site using model-based clustering”, in *Proc. of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM Press, 2000, pp. 280-284.

[4] E. Herder and H. Weinreich, “Interactive Web Usage Mining with the Navigation Visualizer”, in *Proc. of the CHI 2005 Conference on Human factors in computing systems,* ACM Press, 2005, pp. 1451 – 1454.

[5] K. Börner and S. Penumarthy, “Social Diffusion Patterns in three-dimensional virtual worlds”, *Information Visualization*, vol. 2, pp. 182-198, 2003.

[6] N. Hoobler, G. Humhpreys, and M. Agrawala, “Visualizing Competitive Behaviors in Multi-User Virtual Environments”, in *Proc. of the IEEE Visualization Conference 2004* , IEEE Press, 2004, pp. 163-170.

[7] L. Chittaro and L. Ieronutti, “A Visual Tool for Tracing Behaviors of Users in Virtual Environments”, in *Proc. of the 7th International Conference on Advanced Visual Interfaces*, ACM Press, 2004, pp. 40-47.

[8] L. Ieronutti, R. Ranon, and L. Chittaro, “High-Level Visualization of Users’ Navigation in Virtual Environments”, in *Proc. of the 10th IFIP International Conference on Human-Computer Interaction*, Lecture Notes in Computer Science, vol. 3585, Springer Verlag, 2005, pp. 873-885.

[9] ActiveWorlds web site. Available at http://www.activeworlds.com/, last access on March 2006.

[10] VRML, ISO/IEC 14772-2:2004, available at http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772-IS-VRML97WithAmendment1/#ISO_IEC_14772Part2, last access on March 2006.

[11] X3D, ISO/IEC FDIS 19775, available at http://www.web3d.org/x3d/specifications/index.html, last access on March 2006.

[12] J. J. Kuffner, “Goal-Directed Navigation for Animated Characters Using Real-Time Path Planning and Control”, in *Proc. of the International Workshop on Modelling and Motion Capture Techniques for Virtual Environments,* Lecture Notes in Computer Science, vol. 1537, Springer-Verlag, 1998, pp. 171-186.

[13] L. Ieronutti, R. Ranon, and L. Chittaro, “Automatic Derivation of Electronic Maps from X3D/VRML Worlds”, in *Proc. of the Ninth International Conference on 3D Web Technology*, ACM Press, 2004, pp. 61-70.

[14] L. Chittaro, L. Ieronutti, and R. Ranon, “Navigating 3D Virtual Environments by Following Embodied Agents: a Proposal and its Informal Evaluation on a Virtual Museum Application”, *Psychology Journal*, vol. 2, no. 1, pp. 24-42, 2004.

[15] T. Hägerstrand, “What about people in regional science?”, *Papers of the Regional Science Association*, vol. 24, pp. 7-21, 1970.

[16] H. J. Miller, “A Measurement Theory for Time Geography”, *Geographical Analysis*, vol. 37, pp. 17-45, 2005.

[17] R.S. Laramee, H. Hauser, H. Doleisch, B. Vrolijk, F.H. Post, and D. Weiskopf, “The state of the art in flow visualization: Dense and texture-based techniques*”, Computer Graphics Forum*, vol. 23, no.2, pp. 203-222, 2004.

[18] C.P. Risquet and H. Theisel, “Enhanced Integrate and Draw with Feature Visualization”, *Rostocker Informatik-Berichte*, vol. 22, pp. 53-65, 1998.

[19] P. Schulze-Wollgast, C. Tominski, H. Schumann, “Enhancing Visual Exploration by Appropriate Color Coding”, in *Proc. of the 13th Internat. Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2005, pp. 203-210.

[20] Udine3D Web site. Available at http://udine3d.uniud.it (last access on March 2006).