Geo-Aware Process Mining

Carl Corea[1], Patrick Delfmann[1]

[1]Institute for Information Systems Research, University of Koblenz, Germany

Abstract

In the field of physics, a trajectory is defined as the path that an object in motion follows through space. In process mining, trajectories are also studied, however, mostly in a control-flow sense. In this work, we are interested in understanding process trajectories in a *physical* sense, i.e., considering not only the control-flow but also the actual (physical) path that instances followed. To this aim, we present a tool which can visualize directly-follows graphs (DFGs) in a 3-dimensional space, where the placement of the DFG nodes in the space reflects the actual locations of the underlying activities in the real world (e.g., based on geo-location data from the event log). This can be useful for understanding the true movement of process instances in a geospatial manner, for example, where exactly employees are moving within the company building. We describe our browser-based tool and provide pointers for its maturity.

Keywords

Geo-Aware Process Mining, Physical Trajectories, 3D Visualization

1. Introduction

A central objective in process discovery is to understand the *trajectories* of process instances within organizations. For example, in [1], those authors investigate the trajectories of patients in a hospital process.

When speaking of such trajectories, these are often referred to in a control-flow sense. However, there might well be use-cases where one needs to find out not only *how* the instances move, but also *where* the instances are moving (physically). For example, in a hospital process, one may need to check whether wheelchair patients are forced to cover long distances or inaccessible stairways, or whether there are certain locations in the hospital that can become too overcrowded. This is referred to as *geospatial* information.

In this work, we present a tool that can visualize activities in a 3-dimensional space under a consideration of activity-location data (e.g., stemming from event logs). In this way, the x, y, z positions of activities within the visualization reflect the *actual* positions of the underlying activities in the real world. Intuitively, we say that such a form of visualization is *geo-aware*.

An example of such a geo-aware visualization taken from our tool is shown in Figure 1. The use-cases for such geospatial analytics capabilities are many, for example, understanding process movements within companies or hospitals [1], understanding crowd movements in large-scale sport events and touristic regions like ski resorts [2], or understanding movement patterns of animals such as bees [3] or pigs [4].

In the following, we introduce our approach and tool. We will begin with a small background on 3D visualizations, geo-awareness, and our contributions w.r.t. related works.

Proceedings of the Best BPM Dissertation Award, Doctoral Consortium, and Demonstrations & Resources Forum co-located with 22nd Int. Conference on Business Process Management (BPM 2024), Krakow, Poland, September 1st to 6th, 2024.
ccorea@uni-koblenz.de (C. Corea); delfmann@uni-koblenz.de (P. Delfmann)

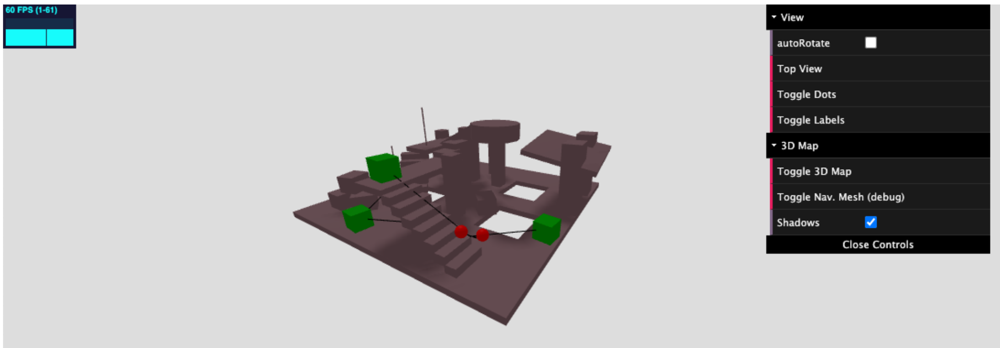


Figure 1: Exemplary geo-aware DFG visualization from our tool (DFG spans across multiple floors).

2. Background

2.1. 3D Visualizations, Geo-Awareness

The Oxford dictionary defines *3D* as *"having, or appearing to have, length, width and depth"* [5]. In this context, an interesting phenomenon about *3D visualizations* is that they are confined to a 2-dimensional image or screen, but the human is able to observe 3D shapes from patterns of light that reflect onto the retina [6]. How exactly 3D perception works is an ongoing research question in fields such as physics or neuroscience, and some explanations involve relationships between patterns of light and physical structures [6]. For this work, we apply the above definition that a 3D visualization is any visualization that appears to have three dimensions.

A prominent example of 3D visualization technology is WebGL [7], which is used for this project. WebGL allows to internally represent 3D graphics. Then, w.r.t. to a viewpoint, WebGL can rasterize the 3D graphics into a 2D projection using shading techniques, which creates the appearance of the shown (2D) image being 3D [7].

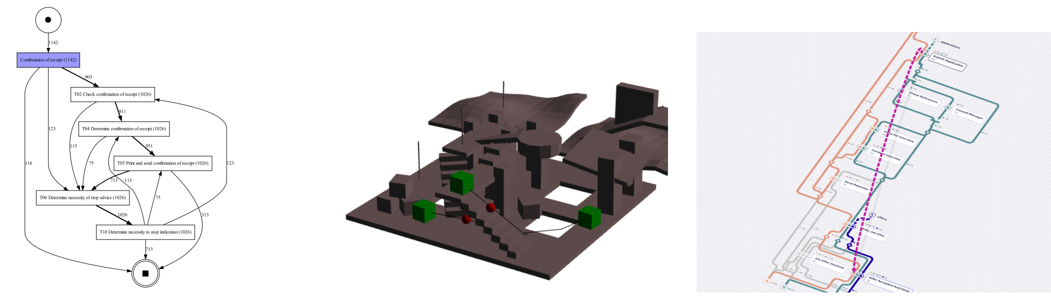
In the scope of 3D process visualizations, we add the constraint that the object placement in the 3-dimensional space should be based on the real-life locations on the objects, leading to, what we call, *geo-aware* visualizations. We define geo-awareness as follows: Consider a directly follows graph $G = (V, E)$; where V is the set of activities, and $E \subseteq V \times V$. Assuming every underlying activity can be pin-pointed to a geographic location in the real world, we assign this location to an activity $v \in V$ via the real-world *latitude*, *longitude* and *height above sea-level*, denoted $v.lat$, $v.lon$ and $v.h$. Furthermore, for any representation R of G (in 3-dimensional space), every node $v \in V$ has an x, y and z position in this space, denoted $v.x_R$, $v.y_R$ and $v.z_R$. We now say that the representation R of G is geo-aware, if it satisfies the following property:

Geo-Awareness. Let $G = (V, E)$ be a directed graph and R be a representation of G. Then, for all pairs of nodes $v, v' \in V$:

- if $v.lat > v'.lat$ then $v.x_R > v'.x_R$
- if $v.lon > v'.lon$ then $v.z_R > v'.z_R$
- if $v.h > v'.h$ then $v.y_R > v'.y_R$

In other words, a representation is geo-aware if the placement of nodes reflects the location of the activities in the real world (and their locations relative to each other). Figure 2 (a) shows a

DFG produced by the PM4PY library. Clearly, such a representation is not geo-aware. For this work (b), novel graph drawing techniques are presented to transform geo-locations into x,y,z coordinates. The novelty w.r.t. other “3D" tools is also clarified in Figure 2 (c), which shows the Celonis Process Sphere. The Process Sphere is “3D", but not geo-aware, as the placement of the nodes is still based on control-flow and does not reflect the real-world activity locations.



(a) Image of a simple DFG: Not geo-aware (node placement based on control-flow).

(b) This work: 3D and geo-aware (node positions are based on the geographical positions).

(c) Process Sphere by Celonis: 3D, but not geo-aware (node placement based on control-flow).

Figure 2: Clarification of geo-awareness (images (a) and (c) taken from PM4PY, and Celonis, resp.).

2.2. Related Work and Motivation

To better show the contributions of our work, we define some properties in the following. First, this work focuses on 3D visualizations (**3D**) that are also geo-aware (**GA**). But of course, many “2D"-approaches are also geo-aware in a 2-dimensional sense. So we define a 2D-version of GA (**GA (2D)**, analogous to GA without y dimension). Second, the work at hand takes a different approach for drawing edges. Most related works work with some form of layers. These layers are however mostly agnostic of each other. As a result, edges drawn on one layer cannot consider the map semantics on a lower background layer, which leads to problems such that edges go “through walls". We argue that—in order to truly understand trajectories— edges should also reflect the actual geographic paths in the real-world. Therefore, our tool works in a single (3D) environment, which allows to apply path-finding algorithms in the 3D world for drawing edges. This results in edges correctly following the maps semantics, e.g., edges going around corners instead of “through" walls. We refer to this property as path-finding (**PF**).

Based on the identified properties, the work at hand compares to other related works for geospatial process visualizations as shown in Table 1. As can be seen, this work is the first to satisfy 3D geo-awarness and path finding.[1] We continue to present our approach.

3. Approach Overview

Our browser-based tool allows to create geo-aware process visualizations. It was built using Three.js. The *tool* and a *screencast* can be found at http://gapm.process-science.uni-koblenz.de.

[1]We acknowledge that [8] is the only work to also actually integrate a cooking recipe in their work, which can also be seen as an important feature.

Tool	3D	GA	GA(2D)	PF
Tiramisù [8]	o	-	x	-
Celonis P. Sphere [9]	x	-	-	n/a
ProM (Plugin) [10]	-	-	x	-
VR-ProcessMine [11]	x	-	-	n/a
BupaR [12]	-	-	x	-
This work	**x**	**x**	**x**	**x**

Table 1: Features of this work w.r.t. related works.

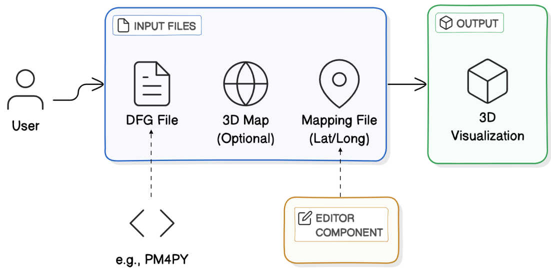


Figure 3: Tool architecture.

The architecture of the tool is shown in Figure 3. The tool takes as input a DFG file (directly follows graph), a mapping file (mapping activities to geo-locations), and (optionally) a 3D map (e.g., created in Blender). The tool is equipped with an interactive editor to create the mapping file in a user-friendly manner. An interactive 3D visualization will then be created based on the input files, by extracting node/edge information from the DFG, and drawing the corresponding graph infused by the provided location data of the nodes. The tool can be run in two modes, depending on whether a 3D map was provided. In the following, we demonstrate both modes.

Mode 1 (with 3D map). For mode 1, a 3D model of the process environment has to be provided. This can e.g. be a map of a building, but also a geographic map of countries. Many open-source tools like Blender exist to create such models. For mode 1, the mapping file assigns to every activity an x, y, z coordinate on the 3D model, and the visualization will place the DFG nodes accordingly. As discussed, an important feature is the integration of path-finding algorithms, which ensures that the edges correctly follow actual paths in the 3D map, and don't go "through walls". An overview of how inputs are combined in mode 1 is shown in Figure 4.

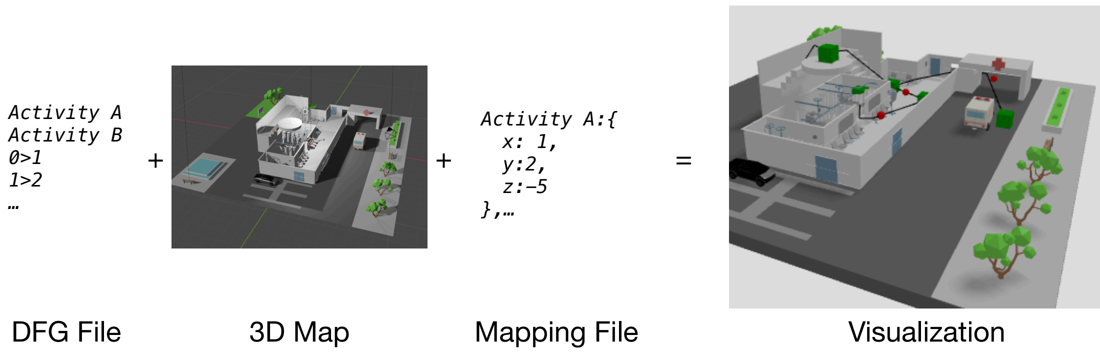


Figure 4: Overview of how inputs are combined in the presented visualization tool.

Mode 2 (without 3D map, positions loaded from geo-data). For mode 2 (without 3D map), the mapping file assigns to every activity a real-life longitude φ, latitude θ, and a height above sea-level h. A remark here is that lat/lon coordinates model positions in a non-euclidean space (i.e., a sphere), however, for our tool, visualizations are presented on a "flat" surface which is a euclidean space. To transform spherical coordinates to cartesian coordinates, we apply the mercator projection. Specifically, for the earth's radius R, we apply the projection $M(\varphi, \theta, h) = (R * rad(\varphi), R * ln(tan(\frac{\pi}{4} + \frac{rad(\theta)}{2})), h)$ [13]. In result, for mode 2 (without map), the nodes extracted from the DFG are placed based on the provided lat/lon coordinates. The camera in the used Three.js library is then set to orbit around the centroidof all nodes. An example of a visualization with positions loaded from lat/lon data is shown in Figure 5.

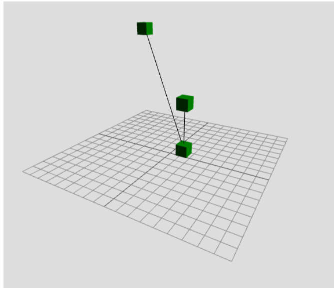


Figure 5: Node positions via lat/lon coordinates.

4. Discussion

The tool can be used with any DFG in the .dfg format. Some considerations are as follows. First, regarding 3D process visualizations, there is a lack of research on quality criteria. For example, crossing lines may need to be assessed differently in 3D spaces. Future works should therefore investigate cognitive aspects of 3D process visualizations to identify quality metrics. Second, as stated, the considered lat/lon coordinates are spherical, but the visualizations are transformed to a euclidean space. For most use-cases, e.g., processes in a building, this will frankly not be of interest to the user, but for larger distances, it should be kept in mind that this will cause distortion, e.g., showing countries on opposites side of the "globe" on one plane. Last, we currently assume a 1:1 mapping between activities and geographic positions, i.e., even across different instances. We argue this is applicable to many real-life settings, e.g., sensory-data from IoT sensors, or machine/warehouse activities. In future works, we aim to integrate support for activities with multiple locations over different instances.

In general, the form of 3D visualizations as introduced in this work can give rise to many future forms of process intelligence, e.g., conformance checking with geospatial reasoning.

References

[1] F. Mannhardt, D. Blinde, Analyzing the trajectories of patients with sepsis using process mining, in: RADAR+ EMISA, CEUR-ws. org, 2017, pp. 72–80.

[2] J. Brunk, D. M. Riehle, P. Delfmann, Prediction of customer movements in large tourism industries by the means of process mining, in: 26th European Conference on Information Systems, 2018, p. 40.

[3] Z. Ahmadi, Y. Bertrand, M. I. Pozo Romero, E. Serral, Analysing the foraging behaviour of bees using process mining: A case study, in: International Conference on Process Mining, Springer, 2023, pp. 5–18.

[4] A. Melfsen, A. Lepsien, J. Bosselmann, A. Koschmider, E. Hartung, Describing behavior sequences of fattening pigs using process mining on video data and automated pig behavior recognition, Agriculture 13 (2023) 1639.

[5] https://www.oxfordlearnersdictionaries.com/definition/english/3d, Accessed: 2024-06-24.

[6] J. T. Todd, The visual perception of 3d shape, Trends in cognitive sciences 8 (2004) 115–121.

[7] https://www.youtube.com/watch?v=f-9LEoYYvE4, Accessed: 2024-06-24.

[8] A. Alman, A. Arleo, I. Beerepoot, A. Burattin, C. Di Ciccio, M. Resinas, Tiramisù: Making sense of multi-faceted process information through time and space (2024).

[9] W. van der Aalst, Object-centric pm: Unraveling the fabric of real processes, Mathematics 11 (2023) 2691.

[10] W. van der Aalst, M. de Leoni, A. ter Hofstede, Process mining and visual analytics: Breathing life into business process models, BPM Center Report BPM-11-15, BPMcenter. org 17 (2011) 699–730.

[11] R. Oberhauser, Vr-processmine: immersive process mining visualization and analysis in virtual reality, in: Proceedings of the 14th Int. Conference on Information, Process, and Knowledge Management, 2022, pp. 75–80.

[12] https://bupaverse.github.io/processanimateR/reference/renderer_leaflet.html, Accessed: 2024-08-01.

[13] M. McClure, Map projection an intro for multivariable calculus, Accessed: 2024-06-24.