Worker-Centric Path Planning: Simulating Hazardous Energy to Control Construction Safety Using Graph Theory

Kilian Speiser and Jochen Teizer

Department for Civil and Mechanical Engineering, Technical University of Denmark

kilsp@dtu.dk, teizerj@dtu.dk

Abstract

Occupational accidents in the European Union continue to pose a significant threat to construction workers, with skill-based errors contributing substantially to these incidents. Virtual training gains prominence in improving skills, but evaluating trainee performance based on safety behavior is challenging to quantify. This paper introduces a pioneering worker-centric simulation that assesses the hazardous energy that a worker may be exposed to while navigating a construction site. The result of the simulation is a Safety Graph, aiding in determining the safest routes for workers. The graph is generated based on a known construction site geometry, with each node representing a one-square-meter area. The simulation developed in the game engine Unity calculates hazardous energy associated with falls and trips that a worker is exposed to when moving between nodes. The evaluation demonstrates a 97% accuracy in estimating hazardous energy. A practical application in virtual training demonstrates how the approach allows for quantifying the safety performances of workers. The study, however, reveals minor shortcomings in the simulation, such as considering an energy threshold or incorporating more hazard types. The results also indicate further applications of the Safety Graph, hinting at its potential in hazard detection or forwarding the safest paths to construction workers using smart glasses.

Keywords –

Occupational Construction Safety, Health, and Well-being; Hazard Simulation; Safe Path Planning; Alternative Routes; Game Engine; Graph Theory; Virtual Education and Training.

1 Introduction

Occupational accidents remain high in the European Union (EU). Approximately 6% of construction workers in the EU face accidents each year [1]. These numbers indicate that, on average, a construction worker undergoes 2.4 accidents in a 40-year career. While various factors contribute to the occurrence of incidents within the construction industry, it is noteworthy that a significant proportion of these incidents are attributed to skill-based errors made by workers [2].

Efficient training is a crucial method to improve the skills of the workforce, with virtual training methods gaining more interest over the last years. Several studies propose virtual training to improve hazard identification [3], train tool handling [4], or collaboration with equipment based on real-world location data [5]. However, the assessment of the trainees in virtual training experiences is challenging. While some studies rely on manual and subjective evaluation methods [6], others analyze collected data [7–9]. Nevertheless, it is challenging to compare the performance of trainees and assess the collected data in a meaningful manner [6,8].

For instance, three workers navigate a construction site: Worker A chooses a safe but longer path, Worker B takes the shorter path with several minor hazards on the way, and Worker C chooses a path with one high-energy hazard that would most likely result in a fatal accident. While subjective evaluation may somewhat evaluate the paths and conclude that Worker A performed the best, an objective data-driven assessment is quantifiable and unbiased. Nevertheless, we would not know if any worker chose the safest available path. To facilitate such assessment, path planning algorithms may allow safety trainers to compare a worker's chosen route with the optimal route. Such optimal routes can be determined using path-planning algorithms.

Path planning is widely adopted in various industries. With the emergence of robots and autonomous vehicles, path planning has also become more relevant in construction. Path planning algorithms can utilize graph theory to find a feasible or optimal route from a starting point to a target. Graph theory is a mathematical approach to model relationships between entities. A graph comprises a set of nodes and a set of edges connecting these nodes [10]. Nodes and edges can represent a wide range of entities and relationships in various fields. Among others, graph theory has been applied to solve assignment problems, transportation problems, knowledge representation, or path planning [10]. To find the optimal route, the edges of the graph, whether directed or undirected, connect the nodes with associated costs. The cost indicates the expenses occurring when using this specific edge. While there is no constraint as to what such cost may represent, standard measures are the traveling time or distance [10]. The path producing the lowest cost represents the optimal path.

In construction, various studies investigated path planning problems. Several studies plan operation paths for construction vehicles [11–13]. Other studies determine optimal routes for crane operations [14] using Building Information Modelling (BIM) [15,16]. Notably, Hu and Fang include safety aspects [15], while the other studies purely focus on productivity aspects. Fewer studies address workers' path planning. Cheng et al. determine paths using workers' trajectory data [11]. Wang and Qin determine safe paths by assessing fall hazards using BIM models [17]. It is noteworthy that all these studies determine path planning using environmental conditions without addressing decisions made by the humans involved when operating. Kim et al. address this shortcoming by integrating deep reinforcement learning to mimic workers' decision-making processes [18].

Building upon the existing literature, the absence of a worker-centric path-finding approach that focuses explicitly on determining the safest paths becomes evident. No previous approach provided a graph that allows for an objective assessment of chosen paths. Such a solution, however, could be integrated into virtual training, as indicated above, but it also allows for the comparison of safety behavior in real-world settings. Hence, the objective of this research is to create a worker-centric algorithm to generate a *Safety Graph* based on the known geometry of a construction site. The generated graph facilitates the risks for a worker to travel between locations within the construction site using the hazardous energy of two of the four most common accidents in the EU [1], namely falls and trips. This *Safety Graph* contains valuable information and can function as input to determine the safest routes for construction workers when navigating through construction sites.

The path planning simulation will be integrated into a virtual training environment to ease the evaluation of trainees' performances. A brief experiment will showcase this application at the end of this paper. A different practical application of the *Safety Graph* could forward the safest path to a construction worker wearing smart glasses and help them navigate the actual construction site. By the end of this paper, readers will gain insights into a cutting-edge approach that combines graph theory, safety assessment, and path planning, which could be used for several practical applications.

2 Research Method

This study employed a mixed research approach to create and assess the proposed simulation. First, a comprehensive literature review investigated worker-centric path planning in construction. Subsequently, a research gap was identified, and the requirements and goals to address this gap were specified. Based on the requirements, a simulation was developed using the Unity game engine. This simulation generates a *Safety Graph* facilitating hazardous energies for a worker.

The simulation was evaluated in an artificial testing environment with fall and trip hazards to validate the approach. Based on the evaluation, the authors could refine the algorithm. To validate the accuracy of the simulation, the authors compared the results from the simulation to a manually created validation graph. Finally, the resulting graph was integrated into a virtual training environment to demonstrate one application of such a *Safety Graph*.

3 Simulation Generating the Graph

The *Safety Graph* assesses the potential hazards faced by human workers while navigating a construction site. This paper presents an algorithm embedded in a simulation framework designed to create such a graph. The following section first overviews the *Safety Graph* before describing the algorithm for creating the graph.

3.1 Algorithm Creating the Safety Graph

As described before, a graph consists of nodes and edges. To generate the nodes, we distributed the construction site into one-meter squares. Each square represents one node in the *Safety Graph*, and adjacent nodes, which a worker can reach, are connected by edges (see Figure 1a). Each edge has a cost, representing the hazardous energy a worker faces when traveling from one node to another. While this cost should encompass all potential hazards like falling, tripping, being struck by objects, or electrocution, this work only integrates falls and trips as they account for more than 40 percent of accidents in construction [1]. Wang and Qin have considered fall hazards in path planning [17], but no previous work included trip hazards in path planning for workers. The novelty of this work lies in generating the *Safety Graph* by simulating the hazardous energy impact on the worker for each possible movement in the construction site. The resulting graph can then function as the base for finding the safest route in the graph (see the blue graph in Figure 1) using search algorithms like Dijkstra [19]. While Figure 1 depicts the shortest path between a particular start and end node, it is crucial to note that graphs enable finding the shortest path for any combination of start and end nodes [10].

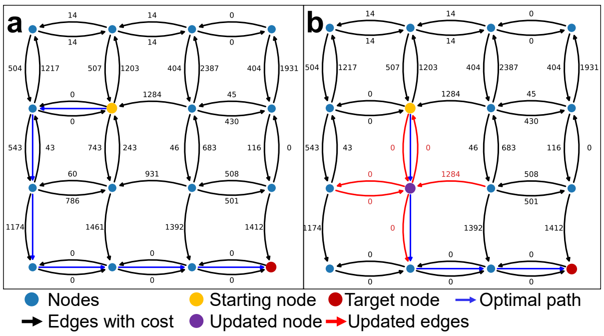

Figure 1. (a) Brief example of a Safety Graph and (b) updating after changes in the construction site.

The benefits of having such a graph are twofold. First, in a static environment, the algorithm runs once. After that, path planning algorithms can efficiently use the graph to identify optimal routes and do not require continuously re-evaluating the site layout. Second, in dynamic environments like construction sites, our graph-based approach allows for updating specific areas of the map without repeating the entire simulation. Figure 1b illustrates the concept: The construction site has changed as, for instance, an object was moved by machinery. Now, the simulations must only update edges connected to this node (highlighted in red in Figure 1b). The update of the node and connecting edges also changes the optimal path from the starting node to the target (see blue edges in Figure 1b).

This section has described the concept of the *Safety Graph* and the motivation behind creating it. The following parts will delve into how our algorithm generates this graph and how the Unity simulation we developed calculates the costs associated with the edges.

3.2 Simulation Development in Unity

The described *Safety Graph* is generated using the algorithm implemented in a Unity simulation. The developed algorithm simulates each possible motion of a worker from one node to an adjacent node and measures the hazardous energy impact on the worker during this motion. For instance, if a worker jumps one meter down during this motion, the hazardous energy of this jump will be considered the cost of this particular movement. Similarly, the energy can be determined for other kinds of hazards such as trips, electrocution, struck-by, or caught-in-between.

The underlying idea of utilizing hazard energy as the edge cost goes back to the findings that the energy of a hazard correlates with the result of an accident [20]. According to Hallowell et al., hazards between 500 Joule and 1,500 Joule are likely to result in medical treatment, while hazards with more than 1,500 Joule most likely cause a severe injury or fatality [21]. The energy intensity would provide more insight, but it would be more challenging to obtain [21]. Thus, in this work, we compute the hazardous energy that a human worker is exposed to when traveling from one location in the construction site to adjacent locations.

To determine the hazardous energy, we developed a simulation framework in the game engine Unity. As Unity facilitates real-world physics, the hazardous energy can be calculated. While the authors expect this algorithm to be generally feasible, obtaining the hazardous energy through the game engine is only one proposed approach. Other methods may utilize camera footage from a site to detect hazards and corresponding safety potential. Nevertheless, this approach has several advantages: It not only includes the safety aspect but also allows for the determination of other parameters, such as accessibility for agents or expected time of traveling.

Listing 1 shows the algorithm to create the *Safety Graph*. The algorithm first distributes the construction site in nodes, adds the agent, and instantiates an empty set of edges (lines 1-3). Then, for each of the nodes, the agent is moved to the neighboring squares. If the motion is possible (no obstacles), the energy potential is calculated, and an edge is added to the graph (lines 5-13). Eventually, the algorithm returns the completed graph in line 14. The following section will first describe the setup of the Unity scene before illustrating the underlying concept of calculating the hazardous energy.

Listing 1. Algorithm generating the *Safety Graph*.

```
1   Function CreateSafetyGraph
2     nodes = SegrateSite()
3     edges = empty collection
4     agent = CreateAgent()
5     For each node in nodes:
6       If edge = SimMotion(agent, node, 1, 0) is not null
7         Then add edge to edges
8       If edge = SimMotion(agent, node, -1, 0) is not null
9         Then add edge to edges
10      If edge = SimMotion(agent, node, 0, 1) is not null
11        Then add edge to edges
12      If edge = SimMotion(agent, node, 0, -1) is not null
13        Then add edge to edges
14    Return Graph G with nodes and edges
```

The simulation undergoes testing in an artificial environment, depicted in Figure 2. This setting comprises several objects that could cause a trip, such as cement bags, a fuel canister, stairs, and a rebar laying area. The objects in the scene were added from different Unity assets. Moreover, the scene features elevated areas from where the worker may fall to a lower level, with some sections safeguarded by guardrails while others remain unprotected. To ensure a thorough evaluation of the algorithm, the authors included several configurations to ensure that the environment is functioning for testing purposes. The algorithm is implemented in Unity, and all related scripts and results are available in a data repository [22].

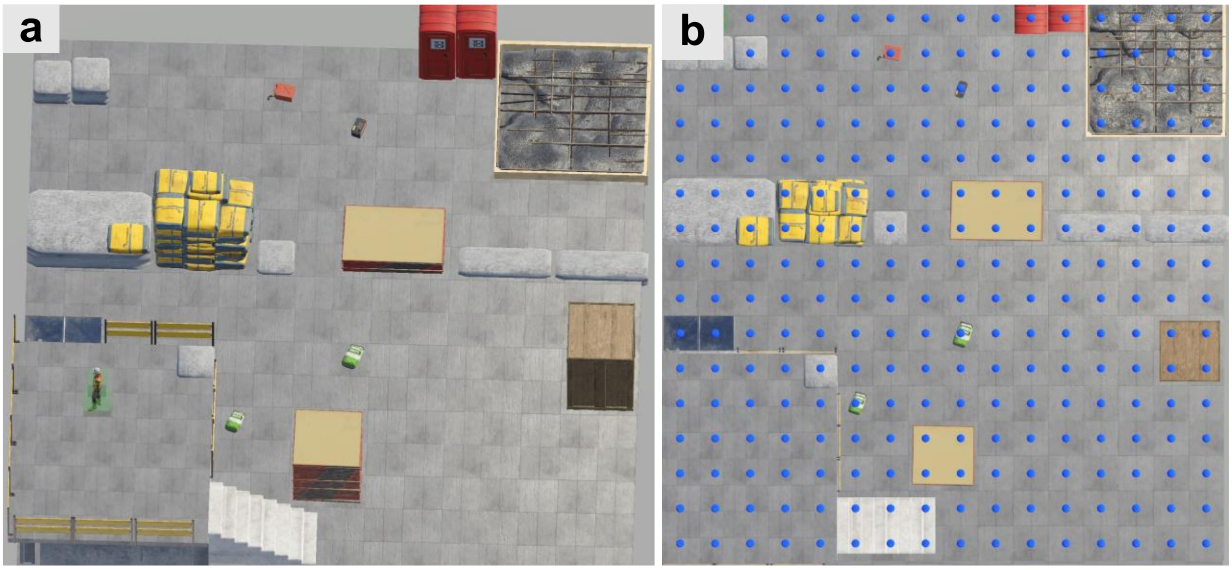

Figure 2. (a) Construction scene in Unity and (b) segregated into nodes (in blue).

3.3 Hazardous Energy Simulation

The proposed algorithm initializes the Graph G, utilizes the geometry of the environment, segregates it into one-meter squares, places a node at each center point (see blue dots in Figure 2b), and adds the nodes to G. After generating the nodes, the algorithm iterates through the nodes and assesses the hazardous energy associated with traveling from each node to its adjacent tiles. As indicated in Listing 1, the simulation of the motions occurs four times for each node. Figure 3 illustrates the process for simulating one motion, and the following paragraphs will describe each of the steps.

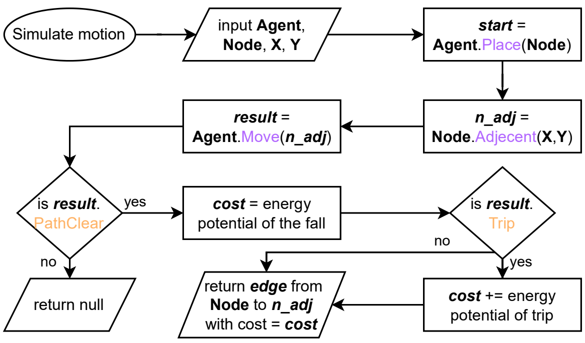

Figure 3. Flow chart for risk assessment of one individual movement of the agent.

The simulation of an individual motion to a neighboring node begins with placing the avatar on the current node n. To ensure the correct vertical placement, we cast a ray downwards from a 100m offset using Unity's *RayCastHit* structure. Then, we place the agent in the position of the first hit.

After placing the avatar, the simulation waits for 100 milliseconds to ensure that the avatar is placed correctly. Then, the algorithm queries the neighboring node n_{adj} from G, and moves the agent towards n_{adj} using the method Rigidbody.MovePosition in Unity. This method moves the agent, but only if the path is free. After moving the agent, the simulation stops for 200ms to evaluate the impact of the motions. For instance, if the worker moves from an unprotected leading edge, the simulation needs to wait until the agent touches the ground. The vertical gravity in Unity is set to -300 to accelerate the simulation.

In case of colliding objects during the movement from n to n_{adj}, the motion will fail, and the simulation continues with the next neighboring node. However, if the motion is successful, our algorithm adds an edge e to the G and calculates the hazardous energy $E_{Hazardous}$ during the motion as the cost of e.

The energy for the fall hazards is computed with Equation 1 where m corresponds to the mass of the worker, g represents the gravity, and Δh corresponds to the difference in height before and after moving the agent.

$$E_{Fall} = m\ x\ g\ x\ \Delta h \tag{1}$$

$$E_{Trip} = \frac{1}{2} m v^2 = \frac{78kg * 10km^2}{2h^2} \approx 300J \tag{2}$$

$$E_{Hazardous} = E_{Fall} + E_{Trip} \tag{3}$$

Equation 2 regulates the hazardous energy for trips, assuming a constant value of 300 Joule. This value corresponds to the kinetic energy of an average human, assuming a mass of 78kg and a velocity of 10km/h. The velocity is selected rather high, integrating a safety factor. The trip hazard is detected using a collider object at the bottom of the worker (see Figure 4). When this object collides with any other object while moving the avatar, the trip hazard energy is added to the cost of e.

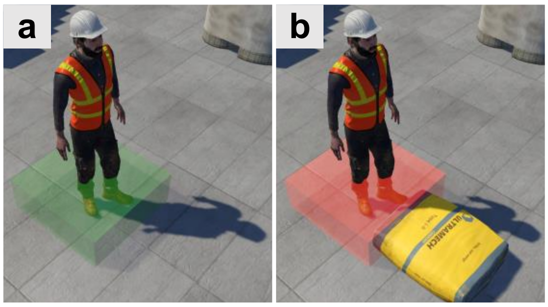

Figure 4. The concept for detecting tripping hazards with (a) a safe path without hazards and (b) a tripping hazard being present.

Figure 5 illustrates the simulation for one node, repeating the previously demonstrated simulation for each adjacent node. There is no edge between node *A* and *B*, as the avatar cannot reach *B* from *A*. Moving to *D* is associated with no risk as there are neither objects causing trips nor an elevation. When moving to *C*, a trip hazard is detected, and the cost of the motion is set to 300 Joule. Lastly, moving to E includes a potential fall of 0.6m (hazardous energy of 400 Joule).

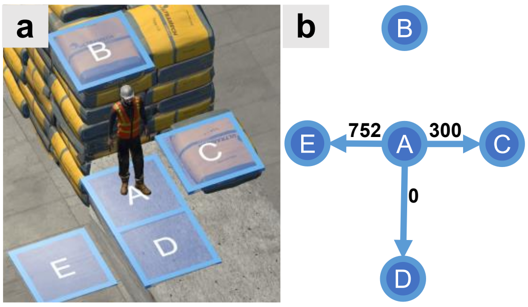

Figure 5. Generating the edges for one node (assuming no diagonal movement is possible).

4 Results and Discussion

The simulation was executed, and the resulting graph was exported into a JSON file for further processing. To validate the approach, the graph was first evaluated regarding its accuracy and then applied in a training scenario with four trainees to find the safest path. Figure 6 depicts the resulting graph highlighting additional information, which the following sections describe and discuss in detail.

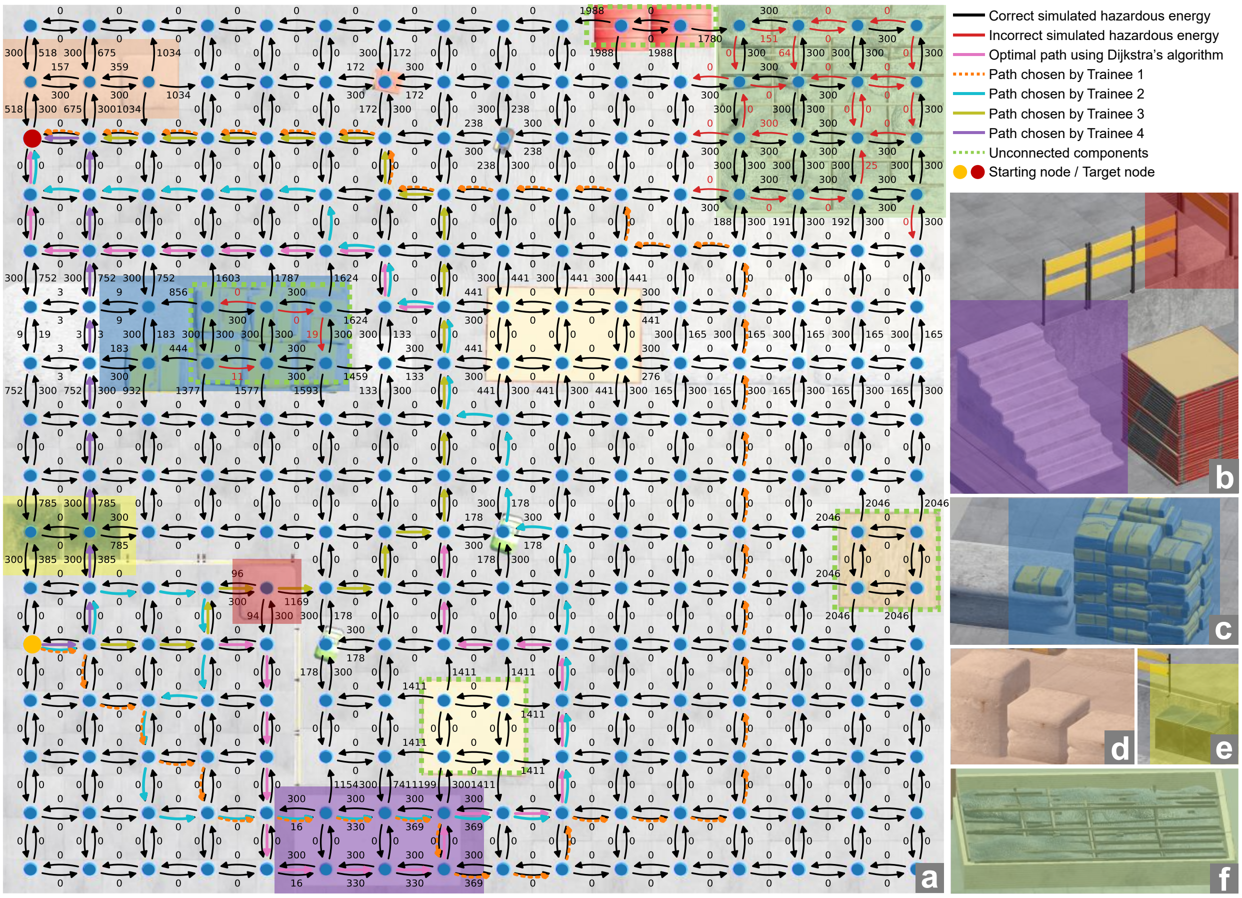

Figure 6. The safety graph includes the hazardous energy from the simulation, critical areas, and the paths taken by the trainees when applying the concept within a virtual training environment.

4.1 Accuracy Evaluation

The simulation's accuracy was evaluated by manually assessing each of the worker's motions. We categorized each of the 1024 edges into one of the five categories shown in Table 1 and created a validation graph (available at [22]). The categories were selected to make sure to detect both falls and trips without losing too much accuracy. Then, we converted the results from the simulation into the same categories based on the hazardous energy. This method allows for the evaluation of the simulation without calculating the exact expected hazardous energy for each of the edges. Still, we manually assessed the exact calculation of the hazardous energy using the area with differently elevated spots shown in Figure 6d.

Table 1. Categorized edges in the validation set.

Cate-gory	Category description	Simulated hazardous energy [J]
0	No hazard	Not accessible
1	*Fall*(Δh < 0,4m)	0 < cost < 300
2	Trip	cost = 300
3	*Fall*(0,4m > Δh > 1m) **or** [*Fall*(Δh < 0.4m)**&***Trip*]	300 < cost < 785
4	*Fall*(1m < Δh < 2m) **or** [*Fall*(1m < Δh 1.6m)**&***Trip*]	785 < cost < 1500
5	High energy hazard	cost > 1500

Using the Python package NetworkX, we compared the edges from the validation graph with the simulation graph. Figure 6a highlights the incorrect simulated edges in red. The comparison of the graphs suggests that the simulation estimates for 97% of the edges the hazardous energy correct. 28 connections are incorrect, of which 25 relate to an underestimate of the hazardous energy and 3 to an overestimate. In 20 cases, the simulation did not detect the expected trip hazard. All these cases occurred in the rebar laying area shown in Figure 6f. Here, the surface is uneven but not uneven enough for the simulation to detect the trip hazard. Similarly, in this area, the simulation detected a trip in five cases, while hazardous energy was expected due to a fall hazard.

Four errors relate to the cement bags in the middle of the scene (Figure 6c), where the surface is also uneven. The hazardous energy while walking on this pile could be considered as a trip or a fall. Nevertheless, it is improbable that a worker would walk up there.

4.2 Safe Path-Planning in Virtual Training

The second step in the validation represents a short case study by integrating the results into a virtual training environment. Four researchers were asked to choose a safe path to navigate from a starting node to a target node in the virtual training environment, as shown in Figure 2. They navigated an avatar in a desktop-based first-person view using a keyboard and mouse (Figure 7). During the training, a Unity script records the trajectory, which we superimpose with the *Safety Graph*. Figure 6a illustrates the chosen paths of the four trainees and the optimal path determined with the Dijkstra algorithm using NetworkX.

Figure 7. The trainee navigates in a desktop-based first-person view using a keyboard and mouse.

Table 2 summarizes the results and suggests that Trainee 4 chose the shortest but most hazardous path. Trainee 4 did not take the stairs but instead jumped down at an unsecured area (Figure 6e) and later crossed the elevated area in Figure 6c. The paths of Trainees 1,2 and 3 result in a similar hazardous energy. However, Trainee 3 chose a shorter path than Trainees 1 and 2. Trainee 3 jumps down at the missing guardrail while Trainees 1 and 3 take the stairs. This situation reveals a shortcoming of our approach. As we aggregate the hazardous energies for each edge, several minor hazards may lead to a worse route than a path with one severe hazard, like Trainee 3. Future improvements should weigh higher energy hazards more than low energy hazards. Another approach would be the removal of edges higher than a certain energy threshold, as the likelihood of an injury is too high.

Table 2. Comparison of the paths of the four trainees.

Trainee	Hazardous Energy [Joule]	Distance [number of edges]	Duration [seconds]
Trainee 1	1,549	41	24
Trainee 2	1,562	37	25
Trainee 3	1,469	22	23
Trainee 4	2,225	11	10
Optimal	1,045	32	-

Another observation relating to the tripping hazards is that Trainees 1 and 2 chose the stairs but later exposed themselves to hazardous energy relating to tripping. In our study, the hazardous energy for tripping hazards is assumed to have a static value. This value should be reconsidered, as a trip close to another hazard could result in a more severe injury. For instance, if the tripping object is at a location where a worker could fall, this edge should be considered with a higher risk.

4.3 Summary

The simulation generally estimates the risk correctly. Further improvements, however, must address uneven surfaces, classify high-energy hazards as unreachable, and consider the secondary impact of detected tripping hazards. In addition, moving elements have not been considered in the current implementation. A potential approach to facilitate dynamic elements is to update the parts of the graph where the changes occurred, as demonstrated in Figure 1.

By connecting the simulation to other data sources, such as a digital twin, the location of other hazard types could be integrated [23]. For instance, detecting hazard zones for struck-by hazards [24] would allow the *Safety Graph* to remove the components within such zones. The simulation would only require the geometry of the construction site from the digital twin. For instance, a BIM model or a point cloud collected by a laser scanner should be sufficient when converted into a mesh. However, further research must validate this expectation.

Additionally, the edges could be accessible to some workers depending on their role. For instance, while the rebar layer should access the area highlighted in Figure 6f, the electrician must not go there. Another approach to declaring such hazard zones and integrating them is the detection based on the actual worker's path [25]. Integrating the trajectory would also allow us to compare how often the workers choose the safe path and help identify tailored training for workers [26].

5 Conclusion and Outlook

While existing studies have explored path planning for construction vehicles and crane operations, few studies focus on workers. A primary contribution of our work is the introduction of a worker-centric path-finding approach, creating *a Safety Graph*.

By representing the construction site as a graph of nodes and edges and assigning costs to edges based on hazardous energy simulations, this research provides a systematic and structured approach to safety evaluation. This novel application of graph theory contributes to a more objective, quantifiable, and data-driven safety analysis and eventually allows for evaluating the safety performance of workers in safety training.

The evaluation of the simulation accuracy, involving the categorization and comparison of edges, adds a layer of robustness to this work. Applying the concept in a virtual training scenario highlights the potential benefits of the approach. Lastly, the study indicates that the *Safety Graph* can demonstrate additional benefits in other applications, such as hazard detection.

Future research can expand the capabilities of the agent by allowing a broader range of motions, thus enhancing the simulation's realism. Additionally, incorporating more types of hazards and consideration of secondary impacts could further refine the graph's ability to assess hazards in dynamic construction environments accurately. Including other agents and dynamic hazards such as a compactor or forklift represents another challenge that future research should investigate. An in-depth study within a virtual training environment is recommended to comprehensively evaluate the impact of this approach on evaluating workers' safety performance. Here, interviews with the trainees could reveal interesting findings on why they chose different routes.

This exploration can contribute to the development of advanced methods for next-generation construction safety training, providing active and personalized feedback to trainees.

Furthermore, testing the proposed approach in a real-world setting, such as forwarding optimal routes to construction workers through an augmented reality (AR) headset, presents an exciting opportunity to validate the practicality and effectiveness of the *Safety Graph*. Such tests would also allow us to investigate the sufficiency of informing workers about the location of hazards or if other measures need to be implemented, e.g., stopping machines or blocking paths.

Acknowledgments

This research has been funded by the European Union Horizon 2020 research and innovation program under grant agreement no. 95398.

References

[1] Eurostat. Accidents at work by NACE Rev. 2 activity and size of enterprise. Https://EcEuropaEu/Eurostat/Databrowser/View/HSW_N2_05/Default/Table?Lang=en, 2023.

[2] Garrett JW, Teizer J. Human Factors Analysis Classification System Relating to Human Error Awareness Taxonomy in Construction Safety. *J Constr Eng Manag*, 135, 2009. https://doi.org/10.1061/(asce)co.1943-7862.0000034.

[3] Wolf M, Teizer J, Wolf B, Bükrü S, Solberg A. Investigating hazard recognition in augmented virtuality for personalized feedback in construction safety education and training. *Advanced Engineering Informatics*, 51, 2022. https://doi.org/10.1016/j.aei.2021.101469.

[4] Bükrü S, Wolf M, Böhm B, König M, Teizer J. Augmented virtuality in construction safety education and training. *EG-ICE 2020 Workshop on Intelligent Computing in Engineering*, 2020.

[5] Speiser K, Hong K, Teizer J. Enhancing the realism of virtual construction safety training: integration of real-time location systems for real-world hazard

simulations. *23rd International Conference on Construction Applications of VR*, 2023. https://doi.org/10.36253/979-12-215-0289-3.15.

[6] Salinas D, Muñoz-La Rivera F, Mora-Serrano J. Critical Analysis of the Evaluation Methods of Extended Reality (XR) Experiences for Construction Safety. *Int J Env Res Public Health*, 19, 2022. https://doi.org/10.3390/ijerph192215272.

[7] Speiser K, Teizer J. An Efficient Approach for Generating Training Environments in Virtual Reality Using a Digital Twin for Construction Safety. *Proceedings of CIBW099W123*, Porto, Portugal, 481-490, 2023. ISBN: 978-972-752-309-2, https://doi.org/10.24840/978-972-752-309-2.

[8] Speiser K, Teizer J. An Ontology-Based Data Model to Create Virtual Training Environments for Construction Safety Using BIM and Digital Twins. *30th European Group for Intelligent Computing in Engineering Conference (EG-ICE)*, 2023.

[9] Golovina O, Kazanci C, Teizer J, König M. Using Serious Games in Virtual Reality for Automated Close Call and Contact Collision Analysis in Construction Safety. *36th International Symposium on Automation and Robotics in Construction*, 2019. https://doi.org/10.22260/ISARC2019/0129.

[10] Bondy JA, Murty USR. Graph Theory with Applications. 5th ed. New York: *Elsevier Science Publishing*, 1982.

[11] Cheng T, Mantripragada U, Teizer J, Vela PA. Automated Trajectory and Path Planning Analysis Based on Ultra Wideband Data. *Comp in Civil Eng*, 26, 2012. https://doi.org/10.1061/(asce)cp.1943-5487.0000115.

[12] Hammad A, Vahdatikhaki F, Zhang C, Mawlana M, Doriani A. Towards the smart construction site: Improving productivity and safety of construction projects using multi-agent systems, real-time simulation and automated machine control. *Proc. of the Winter Simulation Conference*, Berlin, Germany, 2012. https://doi.org/10.1109/WSC.2012.6465160.

[13] Akegawa T et al. Loading an Autonomous Large-Scale Dump Truck: Path Planning Based on Motion Data from Human-Operated Construction Vehicles. *IEEE Intl. Conf. Intelligent Robots & Systems*, 2022. https://doi.org/10.1109/IROS47612.2022.9981828.

[14] Kang SC, Miranda E. Planning and visualization for automated robotic crane erection processes in construction. *Automation in Construction*, 15, 2006. https://doi.org/10.1016/j.autcon.2005.06.008.

[15] Hu S, Fang Y. Automating Crane Lift Path through Integration of BIM and Path Finding Algorithm. *37th ISARC*, 2020. https://doi.org/10.22260/isarc2020/0072.

[16] Lin Z, Petzold F, Hsieh SH. Automatic Tower Crane Lifting Path Planning Based on 4D Building Information Modeling. *Computer Applications - Construction Research Congress,* 2020. https://doi.org/10.1061/9780784482865.089.

[17] Wang TK, Qin C. Integration of BIM, Bayesian belief network, and ant colony algorithm for assessing fall risk and route planning. *Safety and Disaster Management - Selected Papers from the Construction Research Congress 2018*, 2018. https://doi.org/10.1061/9780784481288.021.

[18] Kim M, Ham Y, Koo C, Kim TW. Simulating travel paths of construction site workers via deep reinforcement learning considering their spatial cognition and wayfinding behavior. *Autom Constr*, 147, 2023. https://doi.org/10.1016/j.autcon.2022.104715.

[19] Dijkstra EW. A note on two problems in connexion with graphs. Numer Math (Heidelb) 1959;1. https://doi.org/10.1007/BF01386390.

[20] Alexander D, Hallowell M, Gambatese J. Precursors of Construction Fatalities. II: Predictive Modeling and Empirical Validation. *J Constr Eng Manag*, 143, 2017. https://doi.org/10.1061/(asce)co.1943-7862.0001297.

[21] Hallowell MR, Alexander D, Gambatese JA. Energy-based safety risk assessment: does magnitude and intensity of energy predict injury severity? *Constr Mngmt and Econ*, 35, 2017. https://doi.org/10.1080/01446193.2016.1274418.

[22] Speiser K, Teizer J. SafetyGraph: Worker-Centric Path Planning for Construction Safety Simulating Hazardous Energy [Data set]. DTU Data, 2024. https://doi.org/10.11583/DTU.24754278

[23] Teizer J, Johansen KW, Schultz CL, Speiser K, Hong K, Golovina O. A Digital Twin Model for Advancing Construction Safety. *Construction Logistics, Equipment, and Robotics, CLEaR Conference*, vol. 390, p. 201–12, 2024. https://doi.org/10.1007/978-3-031-44021-2.

[24] Johansen KW, Schultz C, Teizer J. Automated spatiotemporal identification and dissemination of work crews' exposure to struck-by hazards. Proceedings of CIBW099W123, Porto, Portugal: 2023, p. 1–10. ISBN: 978-972-752-309-2, https://doi.org/10.24840/978-972-752-309-2.

[25] Hong K, Teizer J. A data-driven method for hazard zone identification in construction sites with wearable sensors. Proceedings of CIBW099W123, 2023, p. 41–8. ISBN: 978-972-752-309-2, https://doi.org/10.24840/978-972-752-309-2.

[26] Speiser K, Teizer J. Automatic creation of personalised virtual construction safety training using digital twins. *Proceedings of the Institution of Civil Engineers - Management, Procurement and Law*, 2024.