Virtual reality in training artificial intelligence-based systems: a case study of fall detection

Vinh Bui 1 · Alireza Alaei 1

Received: 2 December 2020 / Revised: 31 March 2021 / Accepted: 3 April 2022 / Published online: 14 April 2022
© The Author(s) 2022

Abstract
Artificial Intelligent (AI) systems generally require training data of sufficient quantity and appropriate quality to perform efficiently. However, in many areas, such training data is simply not available or incredibly difficult to acquire. The recent developments in Virtual Reality (VR) have opened a new door for addressing this issue. This paper demonstrates the use of VR for generating training data for AI systems through a case study of human fall detection. Fall detection is a challenging problem in the public healthcare domain. Despite significant efforts devoted to introducing reliable and effective fall detection algorithms and enormous devices developed in the literature, minimal success has been achieved. The lack of recorded fall data and the data quality have been identified as major obstacles. To address this issue, this paper proposes an innovative approach to remove the aforementioned obstacle using VR technology. In this approach, a framework is, first, proposed to generate human fall data in virtual environments. The generated fall data is then tested with state-of-the-art visual-based fall detection algorithms to gauge its effectiveness. The results have indicated that the virtual human fall data generated using the proposed framework have sufficient quality to improve fall detection algorithms. Although the approach is proposed and verified in the context of human fall detection, it is applicable to other computer vision problems in different contexts, including human motion detection/recognition and self-driving vehicles.

Keywords Virtual reality · Training Artificial Intelligence · Fall dataset · Vision-based fall detection

Vinh Bui
vinh.bui@scu.edu.au

Alireza Alaei
ali.alaei@scu.edu.au

1 Faculty of Science and Engineering, Southern Cross University, Gold Coast, Australia
1 Introduction

Machine learning, particularly deep learning and computer vision techniques have been deployed in many domains, including image classification, remote sensing [10] and healthcare [44]. The most important factor for the success of this group of techniques and any generic data-driven learning and classification systems is the availability of a sufficiently large amount of data to train them effectively [16, 20]. However, data, in a sufficient amount and of high quality, is not always available in the healthcare and agedcare domains due to economic and privacy reasons. For example, there are not many high-quality recorded videos of an elderly person’s falls available for public access. The research presented in this paper is undertaken to address such an issue.

Automatic human fall detection has been a hot research topic as of recent years in the health and agedcare sectors as early fall detection is one of the key factors in reducing the severity of fall consequences, especially in elderly people [31, 40, 43, 44]. Researchers have proposed different approaches for fall detection by utilizing the latest development in the field of Internet of Things (IoT), computer vision, and machine learning [24, 43, 52, 54]. However, the lack of real fall data for training machine learning and computer vision techniques has been the main obstacle for achieving the required level of accuracy in these data-driven approaches [14, 25]. Most of the existing algorithms only achieved a good detection accuracy within a limited set of scenarios that they were trained for [43]. To improve the robustness and accuracy of the fall detection algorithms, it is critical to generate a sufficient amount of high-quality fall data, i.e., data that contains representative features of fall and non-fall situations.

Recent advances in computer graphics and Virtual Reality (VR) technology have provided researchers with the capability to simulate real-world environments with high fidelity [7]. VR is used to create realistic and/or futuristic environments in which users emerge and interact. The benefits and usefulness of VR have been demonstrated in many areas, including entertainment, education, tourism, and especially healthcare [13, 32, 42, 44]. An important feature of VR is the ability to set up multiple virtual cameras to record a virtual scene from multiple angles and under different lights and scene settings. Moreover, automatic data annotation can also be done in VR technology. These features make VR technology an excellent platform for simulating human falls and generating high-quality data for automatic fall detection.

This paper, therefore, presents a systematic approach to construct virtual fall datasets while exploring the distinctive features of VR technology. The approach is capable of generating large synthetic fall and non-fall datasets, which are scalable, multi-dimension, maintaining high diversities, and automatically annotated. The generated dataset with the above-mentioned characteristics can satisfy the primary requirements for training machine learning-based fall detection algorithms. In summary, our contributions in this research work are 4-fold: (i) providing a critical review of existing human fall datasets used in automatic fall detection in the literature, (ii) proposing a new human fall data generation framework to overcome limitations of the existing datasets, (iii) generating VR based synthetic human fall datasets with ground truth, and (iv) evaluating the significance of training/pre-training different neural network (NN) fall detection methods using the generated synthetic datasets to obtain more accurate fall detection rates in real scenarios.

The rest of the paper is organized as follows. Section 2 reviews related works in the fall detection domain as well as discusses the use of computer-generated datasets in machine learning and computer vision. Section 3 introduces the proposed method for generating virtual fall datasets. Section 4 details a proof-of-concept case study and the results obtained from the
trained model to evaluate and verify the proposed approach. Finally, Section 5 presents the concluding remarks and future work.

2 Related works

This section summarizes existing techniques used in the fall detection literature and their limitations. It also highlights the needs for computer generated (virtual) fall datasets through a critical review of the existing datasets used in fall detection research. Finally, related works in using synthesized data for training machine learning algorithms are reviewed to reinforce the research presented in this paper.

2.1 Fall detection approaches

Human falls, especially in elderly people, are a major cause of fatal injuries and they can create serious impediments to independent living [31]. According to the World Health Organization, about 28–42% of elderlies aged 65 and older fall every year. Falls are the primary cause of injury-related deaths for this age group [53]. The frequency of falls increases with the age upturn and frailty level [40]. Studies have shown that early fall detection is one of the key factors in reducing the severity of fall consequences [31, 40]. Without assistance, many victims of falls are unable to get up, and the longer periods of lying on the floor can lead to hypothermia, dehydration, bronchopneumonia and pressure sores [40]. This is particularly critical if the person lives alone or loses consciousness after falling.

As shown in recent studies [4, 15, 17, 23, 26, 35, 37–39, 48], it has become very important in the public healthcare domain to develop intelligent and reliable surveillance systems that can automatically track, detect, and timely notify falls. As a result, various techniques have been developed in the literature to automatically detect and prevent human falls [43, 52, 54]. In general, fall detection techniques in the literature can be divided into two main groups: (i) techniques that rely on wearable sensors, and (ii) techniques that are based on computer vision.

Techniques in the first group try to detect falls from abnormal changes in sensor readings. Different types of sensors have been used for this purpose including special health monitoring devices, such as blood pressure and heartbeat sensors, as well as multipurpose devices, including accelerometers and gyroscope in smartphones. Threshold-based and machine learning based classification algorithms have been used to detect and classify if an abnormal change in sensor readings is a fall. A drawback of using wearable sensors is the person under monitoring must wear the sensors at all time. In addition, multiple sensors should be worn to reduce the rate of false positives. As pointed out in [43, 52], the performance of many existing wearable sensor based algorithms was much lower in real-world scenarios compared to what they achieved under a simulated environment.

Unlike the first group, techniques in the second group (computer vision based) detect falls using video footages collected from various types of cameras operating at different frequencies ranging from radio frequency to infrared and visible light. The generic model for fall detection used by techniques in the second group is illustrated in Fig. 1. In this type of model, distinctive features of a fall, such as the magnitude of the acceleration and the angular velocity, are extracted from the video data and then fitted to a classifier to distinguish a fall and a non-fall situation. Machine learning based classifiers are commonly used in this model for feature classification due to their performance in comparison to other classifiers, such as rule-based...
As camera footages contain more contextual information, techniques in the second group often perform better than techniques in the first group in terms of false-positive rate. In contrast, as it takes more time and resource to process video data, the detection time is often longer in the second group.

Although machine learning algorithms have demonstrated superior performance in several fall detection studies [43], to obtain accurate fall detection results from machine learning based approaches, a dataset with a large number of fall data collected from different real scenarios is necessary [25]. However, there is a lack of real-world fall data [43] and collecting a quality real-world dataset is challenging due to cost and privacy reasons. Therefore, it is essential to provide a solution to overcome this data lacking issue. Before presenting a solution to this problem, a review of existing fall datasets is presented in the following subsection to discuss their features and limitations, and then further highlight the need for a VR based fall data generation approach.

2.2 Existing datasets for fall detection

Data play a vital role in training, verifying and testing machine learning, and computer vision algorithms [16, 20]. The performance of these algorithms largely depends on the quality of datasets they were trained with. A comprehensive dataset should have sufficient data quantity and diversity to adequately represent the population of the problem under study. However, real-world datasets, especially in the case of fall datasets, usually are not complete and do not have the required quality.

The literature on fall detection indicates that only a few fall datasets, including the UR Fall dataset [26], the Multiple Cameras (MC) Fall dataset [5], and the Fall Detection (FD) dataset [12], are publicly available and have been used in different research works. An overview of the statistics and characteristics of these datasets is provided in Table 1. Although the datasets contain a substantial set of human fall and non-fall scenarios, they do have some limitations. As shown in Table 1, there is a lack of diversity in the environment, lighting and camera settings. In particular, most of the scenarios in these datasets were recorded from the same camera angles, in the same place, under the same furniture setting and lighting condition. It is well known that application to a new location is one of the critical factors impacting the performance of computer vision and machine learning algorithms, as this type of algorithm generally performs the best under conditions where the training and testing environments are comparably similar [20, 46]. In addition to the diversity issue, the number of fallers is very small with only one faller in each video. The small number of fall events and the dimension of the fall data are other factors that heavily affect the generalization capability of any fall

Fig. 1 The generic fall detection model (adapted from [43])
detection algorithms trained using these datasets. Another limitation of these datasets is that they contain only simulated fall data, i.e., data recorded while the falls were simulated by healthy and young individuals, which may be quite different from the elderly people. However, overcoming these limitations is challenging in practice. Real falls are often not predictable and attempts to record real falls may face privacy issues. Meanwhile, conducting simulated falls in a large number of scenarios is not practical due to time and resource reasons. Hence, generating and collecting data using VR technology, as proposed in this research work, can be a solution to some of the above-mentioned challenges.

2.3 Initial success of virtual datasets and research gaps

The virtual world has been recognized as an environment that facilitates creating online laboratories at a low cost [6]. In the computer vision domain; video games, synthetic images, and 3D modeling were used by researchers as data sources for training various models, including object detection algorithms [22, 41, 49, 50]. For example, crowdsourced 3D CAD models were used to train a Convolutional Neural Network (CNN) for object detection [56]. Synthetic images were used to train a CNN for vehicle detection in [27, 28]. Moreover, video games, such as Half-Life, were utilized to generate a virtual dataset to train an SVM-based algorithm for pedestrian detection in video streams [30]. More recently, the ParallelEye Vision framework, which relies on VR technology, was proposed to generate synthetic natural scenes and virtual images with precise annotations to successfully pre-train a detection model, which was fine-tuned later using real datasets [28]. VR-generated datasets were further used for the development of tree detection/recognition in a driver assistance system [22], and parts recognition in the automated assembly line and production [56]. As a follow-up to their previous work, Li et al. [28] used the ParallelEye-CS virtual dataset for the training and testing of their proposed system for intelligent vehicles.

In the field of fall detection research, there is only a group of researchers who have recently used motion capture technology to simulate human falls. The FUKinect-Fall dataset, which is publicly available [4], contains walking, bending, sitting, squatting, lying and falling actions performed by 21 actors aged between 19 and 72 years old. The FUKinect-Fall dataset is very useful for constructing fall scenarios and can be used to train/test fall detection algorithms.

The initial success of the previous methods in using synthesized data for object detection has motivated us to build virtual fall datasets for human fall detection. To the best of our

| Dataset Features | UR Fall [26] | MC-Fall [5] | FD [12] |
|------------------|--------------|-------------|---------|
| Real/simulated falls | Simulated    | Simulated   | Simulated |
| # Fall scenarios  | 30           | 22          | 143     |
| # No-fall scenarios | 40          | 2           | 79      |
| # Light setting   | 2 × indoor artificial light | 1 × indoor, neon light | 4 × indoor ambient light |
| # Camera / setting | 2 × fixed angle | 8 × fixed angle | 1 × fixed angle |
| # Dimension of data | 3 × Visual, Depth, Accelerator | 1 × Visual | 1 × Visual |
| # Acting Fallers  | 3            | 1           | 9       |
| # Fallers at a time | 1           | 1           | 1       |
| Video format      | 320×240 (30f/s) | 720×480 (30f/s) | 320×240 (25 f/s) |
| Environment       | 2 (Home/Office) rooms | 1 (Lounge) room | 4 (Home/Coffee/Office/Lecture) rooms |
knowledge, there is no existing literature on using virtual reality to generate datasets for fall detection. The proposed VR based data generation method presented in the subsequent section is to bridge this knowledge gap.

3 Methodology

This section details the methodology for generating different fall scenarios. The block diagram of the proposed VR based human fall data generation is depicted in Fig. 2. It starts with the generation of humanoid models, followed by applying different animation methods for simulating the fall models, and finally discussing the construction of the fall datasets. Details of each step are presented in the subsequent subsections.

3.1 Humanoid 3D Model Generation

Although 3D humanoid models are popular in the game and entertainment field, most of them do not have the required quality to simulate human activities realistically for purposes other than gaming and entertainment. Very few research works were conducted on creating 3D humanoid models for biomechanical purposes [1, 8]. In an early work, Boulay et al. [8] have used SimHuman and Mesa library to create a humanoid model with 23 parameters for human posture recognition. Mesa is a 3D graphics library with an API (Application Programming Interface) which is very similar to that of OpenGL. Although this approach is highly flexible for modeling purposes, it requires a substantial coding effort to create a humanoid model. More recently, an adequate procedure has been created by the Make Human (MH) project (MH, 2020) to generate realistic humanoid models mainly used for speech therapy and human anatomy [11]. The MH tool has many functionalities, including built-in functions for creating, calibrating a humanoid model with gender, age, and other biological characteristics. As the models generated by this software are realistic and can be exported to different 3D modeling

![Fig. 2 The proposed human fall data generation framework](image-url)
and VR engines, such as Blender and Unity3D with a choice of geometries, materials, and skeletons information, the MH version 1.1.1 is used to create humanoid models for fall simulation in this research work.

The current version of the software allows choosing 4 skeleton models (or armature) with either 31, 53, 137, or 163 bones. The options with 53, 137 and 163 bones allow capturing finger movement and some facial expressions. As this research work focuses on fall detection and the 31-bone option is more appropriate and sufficient for generating falls from the external motion capture data; the 31-bone option is used in our experimentation. Other options can also be used depending on the motion generation algorithms. For example, the 163-bone option provides a much finer movement control when forward or inverse kinematics are used. Figure 3 shows four different skeleton models and their characteristics of which the 31-bone option is highlighted by the yellow rectangle.

### 3.2 Fall motion generation

In general, a fall motion in VR is generated by animating a humanoid model. The animation is controlled by the model skeleton or armature, which can further be rigged to different poses. Each bone in the skeleton drives a group of vertices of the mass model. The level of bone impact on vertices is set by a set of adjustable weights. Figure 4 demonstrates the impact of the shoulder bone on the mass model using the heatmap. The red area is the place where the highest impact is observed, and it gradually reduces from the yellow to the green areas. The blue area is the place where the bone has no impact on it. Although a fall simulation can be created by manually rigging each bone of the skeleton, this method is labor-expensive and not practical. In the following sub-sections, a few automatic and semi-automatic simulation methods used in our proposed framework for generating synthetic falls are discussed in detail.
3.2.1 Inverse kinematics

The skeleton of a humanoid model can be treated as a set of bones systematically interconnected by joints. The kinematic technique can be used to perform the simulation. Therefore, the kinematic technique, as the first simulation method, is used in our proposed VR based fall data generation. Forward kinematics refers to the process of obtaining the position and angle velocity (direction vector) of an end effector, given the joint variable, i.e., their angles and angular velocities [2]. Figure 5 demonstrates a humanoid arm model with two bones and two joints. The position of the end effector can be computed using the forward kinematic function \( f_k(b_1, a_1, b_2, a_2) \). In the simplest case when the arm is assumed to move in a flat surface, e.g., on a table, the position \((x_e, y_e)\) of the end effector can be computed using the forward kinematic Eqs. (1) and (2):

\[
x_e = b_1 \times \cos(a_1) + b_2 \times \cos(a_1 + a_2)
\]  

\(1\)
\[ y_e = b_1 \times \sin(a_1) + b_2 \times \sin(a_1 + a_2) \] (2)

where \( b_1 \) and \( b_2 \) are the lengths of arm and forearm bones, and \( a_1 \), and \( a_1 + a_2 \) are their angles with respect to the horizontal axes, respectively. The velocity can further be obtained by taking the derivative of \( f(k) \) considering \( a_1, a_2, b_1 \), and \( b_2 \).

Inverse kinematic is the process of finding the corresponding position and velocity of each joint in the system given the position and velocity of the end effector. Inverse kinematic provides the foundation for the automatic creation of a 3D animation, including a fall simulation, as it allows interpolation of the skeleton movements between two poses. However, when the number of bones is greater than two, the inverse kinematic problem is ill-posed and finding a general analytical solution to the problem is difficult [3]. There are several techniques to address this problem, including the Jacobian inverse technique and Heuristic optimization methods [9]. In the context of fall simulation, the inverse kinematic problem can be reformulated into an optimization problem, i.e., finding the optimal position of each bone in the skeleton system to capture a fall motion. However, solving the inverse kinematic optimization problem for fall simulation is a challenging task. Motion capture and machine learning techniques can, however, facilitate solving this challenging problem.

### 3.2.2 Motion capture

Motion capture (mocap) is the process of recording the movements of a real person or object using mocap technologies. The data captured using the mocap technology is then used as constraints to reduce the ambiguity of the inverse kinematic process. As a result, the recorded movements can be reproduced by the humanoid model in a more realistic manner. So, the motion capture is also incorporated, as a simulation method, in our proposed VR based fall data generation framework.

Motion capture technologies can be categorized into two groups: online and offline [34]. The online technologies are often based on magnetic or infrared sensors and their output can directly be used to control a virtual human in real-time to mimic the human performer’s movements. However, the current online mocap technologies have some limitations, including the small number of measurement points, noisy data and cumbersome sensors (although they tend to become smaller). Therefore, the quality of the captured motion largely depends on the data processing software, which handles the data cleaning and data interpolation processes. An inexpensive but rather effective online mocap system of this type is the HTC Vive, which is a full-body tracking system with the VR Mocap software package as shown in Fig. 6. The specification of the latest HTV Vive tracker is provided in Table 2 to shows weight, tracking, dimensions and other characteristics of the HTC Vive Motion Tracker 3.0. The HTC Vive system has successfully been used in the proposed framework and experimental analysis in this research work to capture fall movements.

The offline mocap technologies are mainly based on multiple cameras, which capture optical motions. The cameras track the markers attached to the body of the human performer being tracked. This class of technologies allows the acquisition of subtle gestures to produce high-quality, large and complex movements. However, offline technologies are considerably more expensive compared to their online counterparts, and they require a larger amount of time to process the captured motions. Despite the drawbacks, the offline mocap technologies
are more preferable to capture motions in a clinical context, such as for the assessment of orthopaedical pathologies [36] and obviously for fall simulation. Figure 7 shows a conceptual camera-based mocap system.

Data generated using mocap systems can be used to animate humanoid models. There are several mocap datasets [21, 47], which are publicly available to use for research purposes. The datasets capture common human movements, including walking, running, and jumping. Unfortunately, there are no public mocap datasets for fall motion available at this stage. Therefore, in this research work, a mocap fall dataset was created for fall detection to be publicly available for research purposes.

3.2.3 Physics-based motion generation

To obtain a natural simulated motion, usually, motion-capture data is utilized. However, this approach is limited to those motions that can be replicated and captured. For example, it is difficult to request an elderly person to simulate a fall to capture his/her fall motion. Moreover, the mocap process is time-consuming and may not be quite suitable for the creation of large datasets.

The usage of physics-based simulation has been discussed in actively controlled virtual characters to automate the generation of natural and realistic human motions in an interactive setting without motion data [19]. This technique has been built upon past research [18] that has

| Feature     | Description                                      |
|-------------|--------------------------------------------------|
| Tracking    | Support for SteamVR™ BS1.0 and BS2.0             |
| Weight      | 75 g                                             |
| Dimensions  | 70.9 × 79.0 × 44.1 mm                            |
| Battery life| 7.5 h                                            |
| Field of view| 240°                                             |
included biomechanical constraints into the simulation of animals, to replicate their natural gaits.

The main component of the proposed approach in this research work is based on a biomechanical constraint model containing psychological properties of tendon and muscle fibers. From there, the dynamic of muscle contractions, muscle activation and muscle geometry and skeleton interaction are accurately replicated on the model, providing a natural and accurate muscle contraction and locomotion.

To induce motion, a finite state machine-based control system is used to output muscle excitement signal to control the legs and produce muscle locomotion, based on the muscle dynamic previously modeled. The pose and overall movement of the model are then optimized and improved.

3.2.4 Video-based motion generation

Recently, the usage of uncalibrated videos for synthesizing character movements is rapidly increasing, due to their large scalability and their ability to produce novel motion data. Computer vision and machine learning techniques were used to detect and estimate human poses in video sequences, which were then converted to motion data using some model fitting algorithms [29, 33, 51]. The model is generally the human skeleton and the model variables are the joints. To improve the accuracy, some research works data from inertial measurement units (IMUs) was used. Although this motion generation approach is promising, it is still in an early stage and the quality of generated motions has not reached the required level that can provide a promising accuracy to be used in the healthcare field.

3.3 Fall scenario generation

A fall scenario is composed of at least one fall motion in a virtual environment. The virtual environment can be indoor or outdoor and its settings can be fully customized in terms of surrounding objects and lighting conditions. Virtual fall datasets were generally constructed by shooting and annotating a generated fall scenario from one or multiple virtual cameras with different environment settings and camera angles. Theoretically, any amount of visual data
with high levels of diversity can be generated for training fall detection algorithms. Although other software tools, such as Unreal and Godot, are available in the literature, there is no significant difference between these tools in this fall generation task as they all offer high quality and realistic data. Therefore, in the proposed framework, the Unity3D software is used to generate fall scenarios for experiments and data generation.

It is worth mentioning that Unity3D can simulate real-world physics and as a result, it is possible to simulate other types of motion and directional sensors, such as accelerometers and gyroscope to provide additional data dimensions in the generated fall datasets. These functionalities have not been considered in our fall datasets and this possibility will be explored in our future work.

4 Case study

In this section, a simple case study is presented to verify the usability of the proposed framework for synthetic human fall generation. Our aim in this case study is to evaluate how an existing state-of-the-art fall detection algorithm performs on our synthetic falls. Would the algorithm detect a synthetic fall? What would be the difference in its performance compared to what the algorithm was trained for? To answer these questions, based on the proposed fall data generation framework, three sample fall datasets were first created. A state-of-the-art fall detection algorithm was then chosen to perform prediction using the generated synthetic falls. Finally, the fall detection results are discussed to answer the above-mentioned questions. The following subsections detail the process.

4.1 Virtual fall dataset construction

We used several fall motion generation methods discussed in the previous section to create different sets of data. In particular, we constructed three datasets of fall motions in this case study. The first virtual fall dataset (VFD-1) contains a fall motion created manually using the Blender 2.8 tool. The second (VFD-2) and the third (VFD-3) datasets contain the same fall motion captured using the HTC Vive full-body trackers and the OptiTrack motion capture system, respectively. The reason for using different methods for fall motion construction is that the quality of synthetic falls constructed by each method is different. A manually constructed fall has the lowest quality, while an OptiTrack-captured fall has the highest quality. The synthesized fall motions were then used to drive a humanoid 3D model in a virtual environment to create a fall scenario using the Unity3D simulation engine. The fall scenarios were then captured by virtual cameras to create fall footages, which were used to evaluate fall detection algorithms. Figure 8 shows a photo of the OptiTrack motion capture system used in our experiments.

A summary of the key features of the virtual fall datasets (VFDs) is presented in Table 3. From Table 3 it can be noted that the fall simulation scenarios were conducted in an indoor environment setting. The lighting condition was the default ambient lighting and unchanged during the simulation. Five virtual cameras were used in the simulation to shoot the falls from different angles: two in the front, two from the back and one on the top at 30 fps. Figure 9 shows the camera angles used for the fall simulation. Each fall simulation was carried out for 5 s with two seconds of actual falls. In total, five fall motions were recorded in 300 frames. No-fall scenarios were also generated to test the fall detection algorithms. In the no-fall scenarios,
the fall motion was replaced by a simple walk motion and the rest of the environment settings remained unchanged. As results, The VFDs are composed of 1,200 frames in which five no-fall motions were also recorded.

4.2 Fall detection algorithm

The state-of-the-art fall detection algorithm proposed by Núñez-Marcos et al. [38] was used in this experimental study, as it has shown good performance for fall detection and proven to be one of the best methods in the literature. The fall detection algorithm has been designed based on the VGG16 Convolution Neural Network and the source code is available at https://github.com/AdrianNunez/Fall-Detection-with-CNNs-and-Optical-Flow. Since the algorithm operates on optical flow images, the same Dual TVL1 Optical Flow algorithm proposed by the authors was used to compute the sequence of optical flow images from our generated fall scenarios [55]. The algorithm was selected due to its performance and the availability of the source

Table 3  Virtual fall datasets generated by the proposed framework

| Dataset Features | VFD-1 | VFD-2 | VFD-3 |
|------------------|-------|-------|-------|
| # Falls/frames   | 5/300 | 5/300 | 5/300 |
| # No-falls /frames | 5/1200 | 5/1200 | 5/1200 |
| Environment      | Indoor | Indoor | Indoor |
| # Lighting       | Ambient | Ambient | Ambient |
| # Camera         | 1 × fixed angle | 1 × fixed angle | 1 × fixed angle |
| # Faller         | 1 | 1 | 1 |
| # Dimension of data | 1 × Visual | 1 × Visual | 1 × Visual |

Fig. 8  The OptiTrack motion capture system
codes. The source code of the Dual TVL1 Open Flow algorithm is publicly available on GitHub at https://github.com/vinthony/Dual_TVL1_Optical_Flow.

4.3 Experiment results and discussion

To evaluate the performance of the chosen fall detection algorithm on our datasets (VFDs), we relied on the same evaluation metrics frequently used in the literature [38]. In particular, the algorithm was evaluated using three performance metrics: Precision, Recall, and Accuracy computed using Eqs. (3), (4) and (5), respectively.

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (3)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (4)
\]

\[
\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (5)
\]

In Eqs. (3), (4) and (5), \( TP \) (True Positive) is an optical flow stack labelled as “fall” and was also predicted as fall, \( TN \) (True Negative) denotes an optical flow stack labelled as “no fall” and was predicted as no-fall, \( FP \) (False Positive) is an optical flow stack labelled as “no-fall” and was predicted as fall, and \( FN \) (False Negative) is an optical flow stack labelled as “fall” and was predicted as no-fall. An optical flow stack is a sequence of 10 consecutive frames labelled as “fall” or “no-fall”.

The first experiment was carried out to evaluate the overall detection performance against all synthesized fall footages. The detection algorithm was subsequently presented with

![Fig. 9 The camera angles used in fall simulation](image-url)
footages in VFD-1, VFD-2 and VFD-3 datasets. The obtained experimental results are presented in Table 4. The performances obtained by the algorithm [38] trained and test with three existing datasets (UR-Fall, MC-Fall, FD) are also presented in Table 4 for comparison.

From Table 4, it can be noted that the fall detection algorithm has performed poorly compared to the results and performance claimed by Núñez-Marcos et al. [38]. For example, considering the accuracy obtained using the VFD-3 dataset, the fall detection algorithm has provided an accuracy of 76.6%, which is quite lower than the expected results from the Núñez-Marcos et al. Fall detection algorithm [38]. The accuracy obtained using the VFD-1 is the worst compared to the performance of the fall detection algorithm on the UR-Fall, MC-Fall, and FD datasets. From Table 4, it can further be noted that the Recalls of the fall detection obtained using the VFDs are quite low compared to the fall detection results reported using the UR-Fall, MC-Fall, and FD datasets.

To understand reasons for the inferior performance of the algorithm against VFD datasets and the reason why the fall detection algorithm has performed poorly on VFD datasets, the algorithm was subsequently tested against footage from an individual virtual camera. The evaluation results obtained from the fall detection algorithm using footages from front cameras are shown in Table 5.

From the results shown in Table 5, it is clear that the fall detection algorithm performed significantly better with the image sequences obtained from the two front cameras. Particularly, the fall detection performance obtained using VFD-3 was comparable to the performance achieved with the three existing datasets in all three parameters, i.e., Recall, Precision and Accuracy. This is because the front camera angles were similar to the angles used in the existing fall datasets, with which the detection algorithm was trained. The results obtained from the detection algorithm indicate that the synthetic falls, when recorded from a similar camera angle, have the same quality (of being close to real falls) as the simulated real falls. The experiment also indicates that the detection algorithm failed to detect falls from footages taken from unfamiliar camera angles, e.g., the rear-view or the top-view, as it was not trained with those footages.

| Dataset Performance | VFD-1 | VFD-2 | VFD-3 |
|---------------------|-------|-------|-------|
| Recall              | 16.7% | 23.3% | 36.7% |
| Precision           | 72.5% | 85.0% | 86.7% |
| Accuracy            | 61.3% | 72.7% | 76.7% |

Table 4 Experiment results obtained from the data captured by all cameras

| Dataset Performance | VFD-1 | VFD-2 | VFD-3 |
|---------------------|-------|-------|-------|
| Recall              | 41.7% | 58.3% | 91.75%|
| Precision           | 93.7% | 97.9% | 95.8% |
| Accuracy            | 83.3% | 90.0% | 95.0% |
| # TP                | 5     | 7     | 11    |
| # TN                | 45    | 47    | 46    |
| # FP                | 3     | 1     | 2     |
| # FN                | 7     | 5     | 1     |

Table 5 Experiment results obtained from the data captured by front cameras
In summary, the results obtained from the experiments in this case study allow us to make a few initial conclusions, including (i) the lack of diversity in the training data, particularly the fall motions taken from unfamiliar camera angles, could be the main reason for the poor performance of many fall detection algorithms in real-life scenarios, and (ii) the virtual fall footages have sufficient quality to trigger the fall detection algorithm. This also means that the synthetic data generated by the proposed framework can be used to train fall detection algorithms to improve their detection performance.

5 Conclusion and future work

Fall detection remains a challenging problem in the public health/aged care domain. In this paper, we presented an innovative application of Virtual Reality to address a major obstacle in a real-world fall detection problem— the lack of quality fall data. As a result, VR based fall datasets have been created for training/testing machine learning based fall detection algorithms. The virtual fall datasets will also be made publicly available to researchers for research purposes. The methodology for generating fall data in virtual environments was also discussed and a case study was conducted to verify the quality of the data generated by the proposed approach. The results indicated that the approach can synthesize high-quality fall data, which can potentially be used to improve machine learning based fall detection algorithms.

In future works, this initial research will be expanded in two directions. First, an extensive fall simulation will be conducted using different fall motions captured from both the HTC Vive body motion tracker and the OptiTrack motion capture system to create larger datasets. Second, the generated virtual fall dataset will be used to train/test more fall detection algorithms together with real simulation footages. Other types of sensors, such as accelerators and deep-camera, will further be simulated to provide an additional dimension of fall data for training and testing machine learning based fall detection algorithms. Although the proposed approach was intended to be used for fall detection, it can be applied to other domains, such as in training self-driving vehicles and robotics.

Funding
Open Access funding enabled and organized by CAUL and its Member Institutions.

Data availability
On request.

Code availability
On request.

Declarations

Conflict of interest
The authors declare that they have no conflict of interest.

Open Access
This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.
References

1. Abdel-Malek K, Singh J, A (2013) Human motion simulation: Predictive dynamics. Academic Press, Cambridge
2. Abdel-Malek K, Yang J, Marler T, Beck S, Mathai A, Zhou X, Patrick A, Arora J (2006) Towards a new generation of virtual humans. Int J Hum Factors Model Simul 1(2006):2–39
3. Aristidou A, Lasenby J, Chryssanthou Y, Shamir A (2018) Inverse Kinematics techniques in Computer graphics: a survey. Comput Graph Forum 37(6):35–58
4. Aslan M, Akbulut Y, Şengür A, Ince MC (2017) Skeleton based efficient fall detection. J Fac Eng Archit Gazi Univ 32(4):1025–1034
5. Auvinet E, Rougier C, Meunier J, St-Arnaud A, Rousseau J (2010) Multiple cameras fall dataset. DIRO-Université de Montréal, Tech Rep, 1350
6. Bainbridge WS (2007) The scientific research potential of virtual worlds. Science 317:472–476
7. Bhowmik A (2018) Advances in virtual, augmented, and mixed reality technologies. Inform Disp Arch 34: 18–21
8. Boulay B, Brémond F, Thonnat M (2006) Applying 3D human model in a posture recognition system. Pattern Recognit Lett 27(15):1788–1796
9. Buss SR (2004) Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. IEEE J Robot Autom 17(1–19):16
10. Cai W, Liu B, Wei Z, Li M, Kan J (2021) TARDB-Net: triple-attention guided residual dense and BiLSTM networks for hyperspectral image classification. Multimed Tools Appl. https://doi.org/10.1007/s11042-020-10188-x
11. Cassola VF, de Melo Lima VJ, Kramer R, Khoury HJ (2009) FASH and MASH: female and male adult human phantoms based on polygon mesh surfaces: I. Development of the anatomy. Phys Med Biol 55(1): 133
12. Charfi I, Miteran J, Dubois J, Atri M, Tourki R (2013) Optimized spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and AdaBoost-based classification. J Electron Imaging 22:041106
13. Cipresso P, Riva G (2015) Virtual Reality for Artificial Intelligence: human-centered simulation for social science. Stud Health Technol Inform 219:177–181
14. Cortes C, Jackel LD, Chiang WP (1995) Limits on learning machine accuracy imposed by data quality. In KDD 95:57–62
15. De Falco I, De Pietro G, Sannino G (2019) Evaluation of artificial intelligence techniques for the classification of different activities of daily living and falls. Neural Comput Appl 32:747–758
16. Esteva A, Robicquet A, Ramsundar B, DePristo M, Chou K, Cui C, Corrado G, Thrun S, Dean J (2019) A guide to deep learning in healthcare. Nat Med 25:24–29
17. Feng W, Liu R, Zhu M (2014) Fall detection for elderly person care in a vision-based home surveillance environment using a monocular camera. SIViP 8:1129–1138
18. Geijtenbeek T, Pronost N (2012) Interactive character animation using simulated physics: A state-of-the-art review. Wiley Online Library 31:2492–2515
19. Geijtenbeek T, De V, Der V (2013) Flexible muscle-based locomotion for bipedal creatures. ACM Trans Graphics (TOG) 32:1–11
20. Goodfellow I, Bengio Y, Courville A, Bengio Y (2016) Deep learning 1(2). MIT Press, Cambridge
21. Gross R, Shi J (2001) The CMU motion of body (mobo) database. Citeseer
22. Huang Q (2019) Application of ADAS multi-sensor vision simulation system for tree recognition in urban garden environment. Revista de La Facultad de Agronomia de La Universidad Del Zulia, 36
23. Hussain F, Umair M, Basit, Ehatisham-ul-Haq M, Pires IM, Valente T, Garcia, Nuno M, Pombo N (2019) An efficient machine learning-based elderly fall detection algorithm. arXiv preprint arXiv:1911.11976
24. Igual R, Medrano C, Plaza I (2013) Challenges, issues and trends in fall detection systems. Biomed Eng Online 12:66
25. Khan SS, Hoye J (2017) Review of fall detection techniques: A data availability perspective. Med Eng Phys 39:12–22
26. Kwolek B, Kepski M (2014) Human fall detection on embedded platform using depth maps and wireless accelerometer. Comput Methods Programs Biomed 117:489–501
27. Li X, Wang K, Tian Y, Yan L, Deng F, Wang F-Y(2018) The ParallelEye dataset: A large collection of virtual images for traffic vision research. IEEE Trans Intell Transp Syst 20:2072–2084
28. Li X, Wang Y, Yan L, Wang K, Deng F, Wang F-Y(2019) ParallelEye-CS: A new dataset of synthetic images for testing the visual intelligence of intelligent vehicles. IEEE Trans Veh Technol 68:9619–9631
29. Malleson C, Gilbert A, Trumble M, Collomosse J, Hilton A, Volino M (2017) Real-time full-body motion capture from video and IMUs. 449–457. https://doi.org/10.1109/3DV.2017.00058
30. Marin J, Vázquez D, Gerónimo D, López AM (2010) Learning appearance in virtual scenarios for pedestrian detection. IEEE, 137–144
31. Masud T, Morris RO (2001) Epidemiology of falls. Age Ageing 30:3–7
32. McCarthy C, Uppot R (2019) Advances in virtual and augmented Reality—Exploring the role in health-care education. J Radiol Nurs 38(2):104–105. https://doi.org/10.1016/j.jradnu.2019.01.008
33. Mehta D, Sridhar S, Sotynchenko O, Rhodin H, Shafiei M, Seidel H-P, Xu W, Casas D, Theobalt C (2017) VNect: Real-time 3D human pose estimation with a single RGB camera. ACM Trans Graph 36. https://doi.org/10.1145/3072959.3073596
34. Moeslund TB, Granum E (2001) A survey of computer vision-based human motion capture. Comput Vis Image Underst 81:231–268
35. Mubashir M, Shao L, Seed L (2013) A survey on fall detection: Principles and approaches. Neurocomputing 100:144–152
36. Mündermann L, Corazza S, Andricchi TP (2006) The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications. J Neuroeng Rehabil 3:1–11
37. Noury N, Fleury A, Rumeau P, Bourke AK, Laighin G, Rialle V, Lundy J (2007) Fall detection-principles and methods. IEEE, 1663–1666
38. Núñez-Marcos A, Azkune G, Arganda-Carreras I (2017) Vision-based fall detection with convolutional neural networks. Wireless communications and mobile computing, 2017
39. Özcanhan MH, Utku S, Unluturk MS (2019) Neural network-supported patient-adaptive fall prevention system. Neural Comput Appl 32:9369–9382
40. Peel NM (2011) Epidemiology of falls in older age. Can J Aging/La Revue Canadienne Du Vieillissement 30:7–19
41. Peng X, Sun B, Ali K, Saenko K (2015) Learning deep object detectors from 3d models. 2015 IEEE International Conference on Computer Vision (ICCV), 1278–1286
42. Pillai A, Prabha Susy M (2019) Impact of virtual reality in healthcare: a review. IGI Global, Hershey
43. Ren L, Peng Y (2019) Research of fall detection and fall prevention technologies: A systematic review. IEEE Access 7:77702–77722
44. Robinovitch SN, Feldman F, Yang Y, Schonnop R, Leung PM, Sarraf T, Sims-Gould J, Loughin M (2013) Video capture of the circumstances of falls in elderly people residing in long-term care: an observational study. Lancet 381(9860):47–54
45. Rozantsev A, Lepetit V, Fua P (2015) On rendering synthetic images for training an object detector. Comput Vis Image Underst 137:24–37
46. Schneider S, Greenberg S, Taylor GW, Kremer SC (2020) Three critical factors affecting automated image species recognition performance for camera traps. Ecol Evol 10(7):3503–3517
47. Soomro K, Zamir AR, Shah M (2012) A dataset of 101 human action classes from videos in the wild. Center for Research in Computer Vision, 2
48. Thiao R, Salgado D, Cordeiro M, Osterwald KM, Filho T, Lucena Jr V, Naves E, Murray N (2018) Fall detection system by machine learning framework for public health. Procedia Comput Sci 141:358–365
49. Tian Y, Li X, Wang K, Wang F-Y (2018) Training and testing object detectors with virtual images. IEEE/CAA J Autom Sin 5:539–546
50. Tremblay J, Prakash A, Acuna D, Brophy M, Jampani V, Anil C, To T, Cameracci E, Boochoon S, Birchfield S (2018) Training deep networks with synthetic data: Bridging the reality gap by domain randomization. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 969–977
51. Trouble M, Gilbert A, Hilton A, Collomosse J (2016) Deep convolutional networks for marker-less human pose estimation from multiple views. Proceedings of the 13th European Conference on Visual Media Production (CVMP 2016). https://doi.org/10.1145/2998559.2998565
52. Vallabh P, Malekian R (2018) Fall detection monitoring systems: a comprehensive review. J Ambient Intell Humaniz Comput 9:1809–1833
53. WHO (2015) Falls prevention in older age. World Health Organization. https://extranet.who.int/agefriendlyworld/wp-content/uploads/2014/06/WHO-Global-report-on-falls-prevention-in-older-age.pdf
54. Xu T, Zhou Y, Zhu J (2018) New advances and challenges of fall detection systems: A survey. Appl Sci 8:418
55. Zach C, Pock T, Bischof H (2007) A duality based approach for realtime TV-L1 optical flow. DAMG-Symposium, 214–223. Springer-Verlag, Heidelberg
56. Žídek K, Lazarik P, Pitel J, Hošovský A (2019) An automated training of deep learning networks by 3D virtual models for object recognition. Symmetry 11:496

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.