

Creation of a synthetic dataset for training precise movements of robots for in various industries

Rafael R. Gabdullin¹, Vlada V. Kugurakova^{1*}, Robert T. Iskhakov¹, and Daniil I. Kostyuk¹

¹Kazan Federal University, Institute of Information Technology and Intelligent Systems, 420008, 35 Kremlevskaya st., Kazan, Russia

Abstract. Creating synthetic datasets for artificial intelligence training has a crucial role in modern developments. Considering the difficulties in collecting real data, which is often a costly and time-consuming process that requires significant resources and time. Synthetic data, on the other hand, allows generating large amounts of varied and controlled data that can be customized for specific training and testing needs. This makes the process of algorithm development and improvement more efficient and affordable. This paper presents a comprehensive tool for creating synthetic motion datasets based on rigging a 3D robot model. The ability to create and edit animations through the Blender interface is described. It supports a variety of well-known 3D model formats, providing flexibility in use, and includes powerful tools to achieve high-quality visual effects and realistic scenes. In addition, the tool can automatically generate a large number of robot images needed for training neural networks. By utilizing these capabilities, the tool greatly simplifies the creation of training datasets, making the process more efficient and affordable. Possible future enhancements include automation of rigging, further optimizing the functionality and usability of the tool for robotics and machine learning.

1 Introduction

Modern computer vision, machine learning, and robotics technologies require extensive datasets for training, testing, and operation. However, real-world data collection is complex, costly, and limited in its possibilities. For example, data collection for autonomous vehicles requires recording a large number of miles under different conditions and scenarios, which can be costly and time-consuming. Similarly, data collection for object recognition in complex environments may be limited by the availability of real-world objects or suitable sensors.

In this context, the generation of synthetic data using 3D modelling is a promising approach that provides access to extensive and diverse data for training and testing without the limitations of real data. 3D modelling enables the creation of realistic virtual models of objects and scenes that can be used to generate synthetic data that mimics real data. This

* Corresponding author: vlada.kugurakova@gmail.com

allows researchers and developers to augment limited real-world datasets with virtually unlimited amounts of synthetic data, leading to more robust and generalisable algorithms and models.

Generating synthetic data using 3D modelling has several advantages over real data collection. Firstly, synthetic data allows researchers to generate data in a controlled environment where coefficients, variables and parameters can be changed to study their effects on algorithms and models. Second, synthetic data can be generated in different variations using different variations of the environment, which leads to data diversity and helps to increase the quality of training of artificial intelligence models. Third, synthetic data generation can be generated by automation, making it possible to create large datasets without human involvement.

In this research, a new tool for generating synthetic robot motion data has been developed. The tool is based on accurate 3D modelling and animation to generate high quality datasets that mimic real robotic motion, with the ability to accurately control motion and environmental parameters. Key features of the developed tool include:

- Accurate 3D modelling of the robot, taking into account its moving parts.
- Flexible rigging system for creating realistic motion animations.
- Ability to generate a variety of motion scenarios and interaction with the environment.
- Automatic annotation of generated data to facilitate the training of machine learning models.
- Integration with the popular software Blender to provide extensive visualisation and editing capabilities.

This tool aims to resolve the problem of limited real-world data in the field of robotics and provide researchers and developers with a powerful tool to create large, diverse, and precisely controlled datasets for training robot control algorithms and computer vision systems.

2 Related Works

Before detailing the proposed approach, it is important to refer to existing research in the field of synthetic data generation for robotics and computer vision.

With limited data collection capabilities for training neural networks, the creation and application of synthetic data is not a new problem. In [1], the use of synthetic data and point clouds to create digital twins of autonomous robots is highlighted. Synthetic data enables efficient and cost-effective training in virtual environments, reducing the cost and time associated with real data collection. Synthetic data generated by this method has shown to be competitive with current state-of-the-art methods based on real human motion data, demonstrating scalability to various unseen objects, human motions and can work in different scenarios. In [2], a model is proposed that uses variational autoencoders and generative adversarial networks to capture expressive human movements and transfer these characteristics to robot movements to create more diverse and expressive movements based on different human inputs.

Researchers at Shanghai Jiao Tong University presented a dataset [3] that consists of more than 110000 robot manipulation sequences in different skills and contexts. This dataset includes multimodal perceptual data and is intended to support the development of models capable of generalising skills across different real-world tasks, improving the ability of robots to learn a variety of skills from a single training example.

In [4] different types of synthetic data sources for training autopilot vehicles are compared. It is mentioned that synthetic data tools are evolving to produce increasingly reliable data. Currently, there are a wide range of multi-task datasets that simulate a variety of situations, among which Virtual KITTI [5], VIPER [6], PreSIL [7], etc. are highlighted.

One of the studies [8] mentions the application of existing technologies to create virtual cities using Unity Perception. This tool provides the ability to generate synthetic data with annotations. It supports various computer vision tasks including 2D/3D object detection and semantic segmentation. The package includes a domain randomisation feature that adds diversity to the generated datasets, which increases the generalisability of machine learning models.

One of our previous works [9] was to create datasets of large crowds of people in a building. The feature to create a variety of data was supported by the character module (capable of creating up to 100000 unique models of people including a variety of poses), the object module (capable of changing the position and appearance of surrounding objects on the location), and the camera module with various settings (sensor height and width, focal length, aperture value, etc.).

3 Development of a dataset creation tool

Creating synthetic data for training a neural network requires several important requirements, without which the resulting data will be useless. The synthetic data generator must create new data that mimics real-world observations and be used to train machine learning models based on existing data, applying various generation methods. It should be possible to edit and change various parameters of real-world data to produce a variety of data within a single object. Also, an important requirement is to work with virtual objects, which allows obtaining data not only from real objects, but also from objects created with the help of computer programmes. Finally, annotation and markup of data are necessary for testing and validation of models, which will allow neural networks to better perceive input data and train the model more efficiently, also annotations can be useful for assessing the quality of the model and identifying errors in the future.

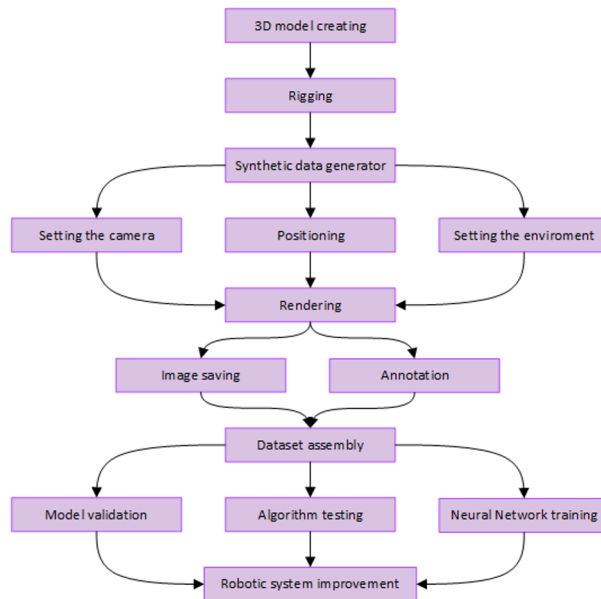


Fig. 1. Algorithm for creating and implementing a synthetic dataset.

The block diagram presented in Figure 1 shows the key steps of the proposed approach to creating and utilising a synthetic dataset for robotic systems. Each of these steps will be discussed in detail in the following sections.

3.1 Creating a 3D model and preparing for animation

One of the first steps in creating a virtual dataset is to create a digital twin of the object in a virtual environment (Figure 2) and set it up for animating moving parts. Creation of the 3D model mesh and its further preparation for animation will be done in the Blender 3D model editor.

In creating low-polygon 3D models, we will consider the following requirements:

- A 3D model of a robot is an accurate copy of a real object with precise dimensions, proportions, rotation and detailing.
- It is necessary to use the optimal (minimum, if possible) number of polygons. This is necessary to reduce the load on the processor and video card. It is important to think carefully about the model topology and avoid excessive detailing.
- Reduce the number of hidden geometric elements. Avoid creating hidden surfaces or geometry that is not visible from certain viewpoints.
- Proper polygon placement. Each polygon must be deployed with the correct side, that is, it must respect the inner and outer sides in the correct position.
- 3D model positioning. The model must be located at the coordinate origin. The scale of all objects must be equal to one, and rotation and location must be zero.

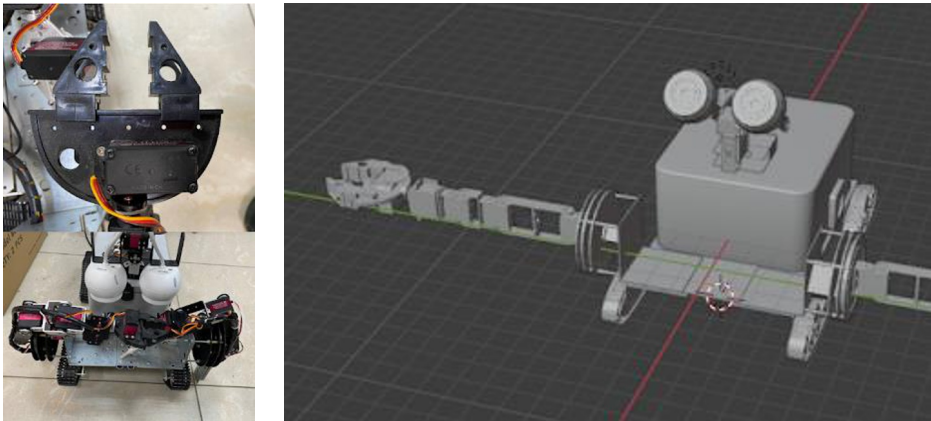


Fig. 2. Actual model of the robot (left) and its virtual replica (right).

Once the 3D model has been created, the animation stage can be proceeded to. Rigging, or the process of animation controllers installing [10], is a stage of preparing a three-dimensional character model for animation. As part of this process, a special set of virtual joints and bones are created and placed inside the pre-created model. This set defines the structure and possibilities of the character's animation, as well as provides for the peculiarities of its movement and transformation.

For this purpose, it is necessary to specify the position of the "bones" and their exact connection, and it is important to take into account the exact snap of the model so that all possible positions of the moving parts, including rotations, are considered. For further programming convenience, all bones were named and combined into one object (Figure 3).

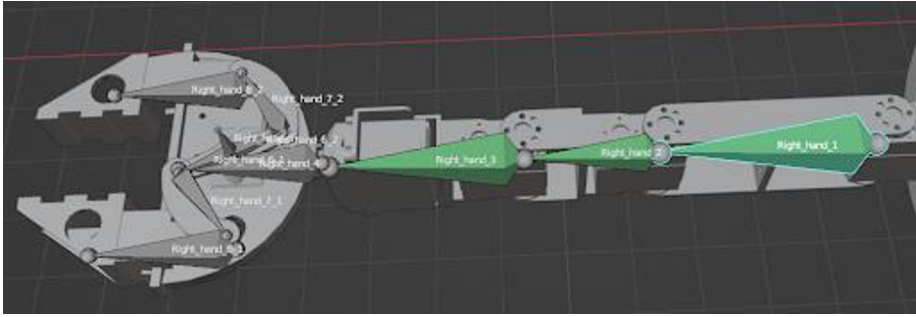


Fig. 3. Bones set along the shape of the model's contour.

In order to avoid the problem of unattainable movements when individual parts of the model pass through the mesh of the 3D model, we need to introduce so-called "bone constraints" to limit the movement of individual bones. In this way we introduce realistic movements and avoid possible errors in the dataset.

It should be noted that an important component of the skeleton is the root bone, which is the basis of the model for linking the other bone chains responsible for the movement of the arms or the upper part of the robot. The main function of the root bone is to be able to track the rotation of the caterpillars and measure the distance travelled by the robot.

3.2 Development of a synthetic data generator

After preparing the 3D model of the robot for animation, the next step was to develop a software component for generating a variety of movements and collecting synthetic data.

One of the advantages of the Blender editor used is the ability to obtain data about the elements of the 3D model and work with them in the Python language using the Blender Python API library. The control of robot poses and virtual cameras of the scene to create frames will be implemented based on this library.

The dataset is created by continuously monitoring and photographing the scene with cameras. During each time step, the cameras capture images and store them in a file system, creating a diverse collection of photos. This process ensures the creation of a dataset consisting of multiple images representing the various movements and actions of the robot in the scene.

In order to ensure ease of use and standardisation of data, all images are saved in PNG format (Figure 4). This format is preferred because of its advantages, such as support for transparency, no loss of image quality during compression, and wide compatibility with various software and hardware platforms.



Fig. 4. Dataset image set.

The collected dataset is an important tool for training and developing machine vision and image processing algorithms, allowing the creation of models capable of adequately perceiving and analysing the environment with the robot.

To improve the motion recognition process, it is proposed to introduce a system for adding accompanying information (Figure 5), such as bone names and rotation angle in radians, etc., to improve the motion recognition process. Such annotation contributes to a more accurate identification of the features of movements and poses of objects in the images when training the neural network.

```
{'left_hand_1': {'x': (0.0, 0.0), 'y': (0.0, 0.0), 'z': (-1.483529806137085, 1.483529806137085)}}
{'left_hand_2': {'x': (0.0, 0.0), 'y': (0.0, 0.0), 'z': (-1.483529806137085, 1.483529806137085)}}
{'left_hand_3': {'x': (0.0, 0.0), 'y': (0.0, 0.0), 'z': (-1.483529806137085, 1.483529806137085)}}
{'left_hand_5_2': {'x': (0.0, 0.0), 'y': (0.0, 0.0), 'z': (-0.3490658402442932, 0.0)}}
{'right_hand_1': {'x': (0.0, 0.0), 'y': (0.0, 0.0), 'z': (-1.483529806137085, 1.483529806137085)}}
{'right_hand_2': {'x': (0.0, 0.0), 'y': (0.0, 0.0), 'z': (-1.483529806137085, 1.483529806137085)}}
{'right_hand_3': {'x': (0.0, 0.0), 'y': (0.0, 0.0), 'z': (-1.483529806137085, 1.483529806137085)}}
{'Head_bone_1': {'x': (0.0, 0.0), 'y': (0.0, 0.0), 'z': (-1.483529806137085, 1.483529806137085)}}
{'Head_bone_2': {'x': (-1.483529806137085, 1.483529806137085), 'y': (0.0, 0.0), 'z': (0.0, 0.0)}}
```

Fig. 5. Information about the range of displacement of the bones.

Having created a tool for generating synthetic data, it is important to consider the potential applications and benefits of using the resulting dataset.

3.3 Dataset usage

One of the perspectives of robot training in a virtual environment is the use of computer vision to automatically determine the position of the robot and its manipulators in space, including the use of a marker system [11]. The use of markers makes it possible to calibrate the position of manipulators in an automated way if necessary. That said, the perspective of using virtual spaces to train artificial neural network to determine the position of the arms without the need for calibration and markers is an interesting and promising approach. Such a method has enough advantages such as no risk of damaging the robot, the ability to easily modify training conditions and scenarios, and the ability to rapidly prototype new approaches, thus being used further in areas such as medical robotics [12]. That said, there are a number of drawbacks associated with the use of a simulation environment that introduce limitations and uncertainties in the application of this method. For example, the need for large computational data for artificial neural network training and the detachment from real data (Sim2Real Gap), which is a great challenge for developers due to the presence of a large degree of difference between the virtual and real robot model. Ignoring this problem will inevitably lead to incorrect training and hence incorrect robot operation. Therefore, this problem is being actively researched and already has an example of a solution [13].

The development of remotely controlled robotic devices using virtual reality (VR) headsets is also quite popular [14, 15]. This type of interaction will be more familiar to the operator, as he or she will be able to follow the robot's movements from a first-person perspective and control the movements of the arms through hand movements. This task requires great accuracy in the robot's response to user interactions, as the transmission of video images to the VR headset camera can lead to seasickness, and errors in controlling the manipulators will not allow the robot to cope with the required task correctly. For example, the use of neural networks will be useful in synchronising the movements of an operator using a VR headset with the movements of the arms of a remote robotic device with a stereo pair. In this case, gestures and positions of controllers [16] can be matched to the generated images of robot motion.

4 Conclusions

The proposed concept implemented in the pilot tools demonstrates a high degree of versatility and adaptability to different typologies of robotic systems. The synthetic data methodology implemented in the tool leads to significant resource optimisation, reducing the time and financial costs associated with the process of empirical data collection.

An important advantage of the tool is the provision of enhanced research safety. The ability to test and train algorithms for controlling robotic systems in a virtual environment minimises the risks of damaging physical equipment. In addition, the tool's functionality enables rapid generation of large diverse data sets, facilitating the creation of large-scale datasets required for training complex machine learning models.

Researchers have precise control over the parameters of the generated data, allowing them to simulate scenarios that are difficult or impossible to replicate in real-world settings. The use of Blender software and the Python API provides a high degree of integration with a wide range of existing tools and libraries for 3D modelling and animation.

Some technical limitations of the developed solution can be noted:

- Lack of functionality for lighting control;
- Loading of any 3D model of the robot is not implemented;
- No functionality for automatic creation of robot skeleton for automatic preparation for animation.

Thus, the developed solution leaves space for future improvements and scalability to work with any robot models. The tool can be enhanced by automatically analysing the geometry and structure of the loaded 3D model to identify potentially moving parts and prepare for animation. This could involve the use of neural networks [10, 17] or pre-recording animations from a real robot instance, for example using motion sensors to further analyse the model and prepare for the creation of synthetic data.

Summarising the results of our study, we can formulate key conclusions and outline the prospects for further development of the developed tool. The developed tool has extensive functionality for working with 3D models and their animations. Users can create and edit animations using a user-friendly interface for precise parameter control. It also provides the option of selecting various templates and presets to simplify working with animations.

Users can also easily customise the appearance and materials of models using powerful third-party texturing and lighting tools. This allows for high quality visual effects and realistic scenes. The presented tool for generating synthetic robot motion data is a powerful tool for creating training datasets. As mentioned above, there is considerable room for improvement: automation of rigging, integration with CAD systems, use of machine learning and improved visualisation - can significantly increase the efficiency and functionality of the tool. These improvements will not only help to speed up the process of creating synthetic data, but also make it more accessible and user-friendly, ultimately leading to better and more accurate models for robotics and machine learning.

Acknowledgments

This paper has been supported by the Kazan Federal University Strategic Academic Leadership Program ("PRIORITY-2030").

References

1. S. Christen, L. Feng, W. Yang, Y.-W. Chao, O. Hilliges, and J. Song, *Synh2r: Synthesizing hand-object motions for learning human-to-robot handovers*. [Online]. Available: <https://arxiv.org/abs/2311.05599> (2023)
2. P. Osorio, R. Sagawa, N. Abe, and G. Venture, *A generative model to embed human expressivity into robot motions*, *Sensors*, **24**, 2. doi: 10.3390/s24020569 (2024)
3. H.-S. Fang et al., *Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot*. [Online]. Available: <https://arxiv.org/abs/2307.00595> (2023)
4. Z. Song et al., *Synthetic datasets for autonomous driving: A survey*, *IEEE Transactions on Intelligent Vehicles*, **9**, 1, pp. 1847–1864. doi: 10.1109/tiv.2023.3331024 (2024)
5. A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, *Virtual worlds as proxy for multi-object tracking analysis*, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4340–4349. doi: 10.1109/CVPR.2016.470 (2016)
6. S. R. Richter, Z. Hayder, and V. Koltun, *Playing for benchmarks*, in 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2232–2241. doi: 10.1109/ICCV.2017.243 (2017)
7. B. Hurl, K. Czarnecki, and S. Waslander, *Precise synthetic image and lidar (presil) dataset for autonomous vehicle perception*, in 2019 IEEE Intelligent Vehicles Symposium (IV), pp. 2522–2529. doi: 10.1109/IVS.2019.8813809 (2019)
8. J. Zhang, T. Fukuda, and N. Yabuki, *Automatic generation of synthetic datasets from a city digital twin for use in the instance segmentation of building facades*, *Journal of Computational Design and Engineering*, **9**, 5, pp. 1737–1755, doi: 10.1093/jcde/qwac086 (2022)
9. V. V. Kugurakova, V. D. Abramov, D. I. Kostiuk, R. A. Sharaeva, R. R. Gazizov, and M. R. Khafizov, *Generation of three-dimensional synthetic datasets*, *Russian Digital Libraries Journal*, **24**, 4, pp. 622–652. doi: <https://doi.org/10.26907/1562-5419-2021-24-4-622-652> (2021)
10. R. Gazizov and A. Shubin, *Procedural methods for skinning humanoid characters*, *Russian Digital Libraries Journal*, **25**, 5, pp. 404–440. doi: <https://doi.org/10.26907/1562-5419-2022-25-5-404-440> (2022)
11. K. Shabalina, A. Sagitov, M. Svinin, and E. Magid, *Comparing fiducial markers performance for a task of a humanoid robot self-calibration of manipulators: A pilot experimental study*, in *Interactive Collaborative Robotics*, in *Interactive Collaborative Robotics*, pp. 249–258. doi: http://dx.doi.org/10.1007/978-3-319-99582-3_26 (2018)
12. B. Abbyasov, A. Zagirov, T. Gamberov, H. Li, and E. Magid, *Vision-based autonomous navigation for medical examination using a ur3e manipulator*, in *Conference on Artificial Life and Robotics*, **29**, pp. 308–311. doi: <https://doi.org/10.5954/ICAROB.2024.OS11-4> (2024)
13. J. Theo, B. Guillaume, V. Romain, and W. Christian, *SIM2REALVIZ: visualizing the sim2real gap in robot ego-pose estimation*, [Online]. Available: <https://arxiv.org/abs/2109.11801> (2021)
14. V. V. Kugurakova et al., *Vr-telecontrol of multi-arm devices: Problems, hypotheses, problem statement*, *Russian Digital Libraries Journal*, **25**, 5, pp. 441–471. doi: <https://doi.org/10.26907/1562-5419-2022-25-5-441-471> (2022).
15. V. Kugurakova, M. Khafizov, S. Kadyrov, and E. Y. Zykov, *Remote control of a robotic device using virtual reality*, *Software & Systems*, **35**, 3, pp. 348–361. doi: <http://dx.doi.org/10.15827/0236-235X.139.348-361> (2022)

16. N. Nikiforov, T. Tsoy, R. Safin, Y. Bai, M. Svinin, and E. Magid, *Pilot studies on avrora unior car-like robot control using gestures*, in Electromechanics and Robotics: Proceedings of 16th International Conference on Electromechanics and Robotics”, St. Petersburg, Russia, 14–17 April 2021, pp. 271–283. doi: https://doi.org/10.1007/978-981-16-2814-6_24 (2022)
17. Z. Xu, Y. Zhou, E. Kalogerakis, C. Landreth, and K. Singh, *Rignet: Neural rigging for articulated characters*, [Online]. Available: <https://arxiv.org/abs/2005.00559> (2020).