Scientific Visualization, 2023, volume 15, number 2, pages 80 - 89, DOI: 10.26583/sv.15.2.07

# Computer Simulation and Visualization of Wheel Tracks on Solid Surfaces in Virtual Environment

A.V. Maltsev[1]

Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences" (SRISA RAS)

[1] ORCID: 0000-0003-1776-814X, avmaltcev@mail.ru

**Abstract**

The paper describes methods and approaches for implementation of traces from virtual prototypes of wheeled technical vehicles on solid objects in three-dimensional virtual scenes. Use of particle systems hierarchically linked to wheel models is the basis of solutions developed. Original type of particle system is proposed, in which generation of elements is regulated by special control signal. Creation, processing, storage and destruction of particles is performed by means of the CUDA architecture with use of computing cores of modern graphics processor and video memory resources. Creating geometric model of each particle and its rendering is carried out on programmable graphics pipeline with own vertex, geometry and fragment shaders. Visualization of virtual environment with simulation of wheel tracks is performed in real time. Software modules were created based on developed methods and approaches. Approbation of them was carried out in virtual environment system VirSim and showed adequacy and effectiveness of proposed solutions for simulation of wheel tracks in virtual environment systems and training complexes.

**Keywords**: visualization, virtual environment, 3D modelling, wheel track, particle system, graphics processor, CUDA.

## 1. Introduction

At present, virtual environment synthesized by a computer is used in many areas of human activity, from developments in the gaming industry to creation of tools to teach qualified specialists [1-6]. Usually it is a three-dimensional scene where known or promising objects of the world around us are modeled, as well as their behavior and interaction. Moreover, the degree of perception of this environment by a person largely depends on the simulation quality and a level of detail of virtual models, which is especially important for such applications as training complexes and virtual environment systems.

An important class of objects transferred from real space into the virtual one is wheeled technical vehicles, for example, cars, construction equipment, loaders, robots, etc. In real life, such objects can leave traces on the surfaces on which they drive, due to the heavy weight or dirty wheels. Therefore, in order to increase the realism of three-dimensional scenes, development of effective methods and approaches for wheel track simulation in virtual environment is an important and actual task.

One of the research directions in the field of creation and development of technologies for wheel track implementation is its simulation by changing virtual surface geometry along which technical vehicle drives. Thus, the authors of paper [7] propose methods for increasing the detail of landscape polygonal mesh and its correction at the areas of interaction with wheels. Articles [8, 9] consider changing height map of virtual terrain in real time based on additional textures generated using the graphics processor (GPU), namely: initial terrain depth texture, vehicle depth texture, depth offset map, and deformed terrain depth texture.

In general, the peculiarity of methods based on landscape correction lies in the possibility of their application for modeling wheel tracks on bulk or ductile materials.

A different approach is demonstrated in [10]. It considers the technology for implementing tire tracks from virtual car model when creating animation by means of the Blender 3D [11] software package. The author's idea is to use particle systems. In this case, each element of the system is a small part of track from wheel with which this system is linked. The technology is promising for simulation of traces on hard surfaces. However, the use of standard particle system with a given element generation frequency leads to irregular distribution of particles along the track (there are areas with absence or excessive accumulation of particles), especially when changing a speed of simulated vehicles.

This paper proposes original methods and approaches for modeling tracks from wheels of technical vehicles on solid surfaces in three-dimensional scenes, including the fact detection of wheel contact with pollutants, characteristics determination of its contact with a surface, as well as creation and visualization of track elements. The novelty of developed solutions is implementation of the tracks based on particle system with controlled generation of its elements, as well as using the capabilities of CUDA architecture to create, parallel processing and destroy particles of wheel tracks. The latter makes it possible to simulate traces of great length in real time.

## 2. Particle system for wheel

To simulate tracks from wheels of technical vehicles on the hard surfaces of virtual scenes, this article proposes an approach that is the development and significant processing of the idea presented in [10]. The main essence of this idea is that each wheel uses its own particle system, textured elements of which fall on the surface in contact with the wheel and together form a track.

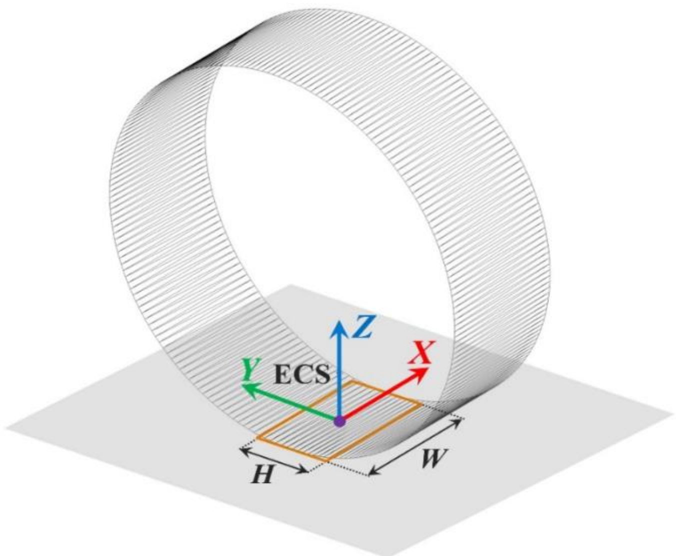


Fig. 1. Placing particle system's emitter on the outer surface of wheel model.

Figure 1 shows proposed scheme for placing a rectangular emitter of particle system on the outer surface of three-dimensional virtual model of a wheel. Note that in considered solution, the emitter has hierarchical link with this model, being its child node. Let the origin of the emitter's right-handed coordinate system ECS be at crossing point of its diagonals, the $Z$-axis is directed perpendicular to its plane, whereas the $X$-axis and $Y$-axis are parallel to its sides. It is generally accepted that particles are generated in the negative direction of the $Z$-axis of the ECS. Because of this, the initial orientation of the emitter must be set so that the positive direction of this axis points to the center of wheel along its radius. The $X$-axis of the ECS system is selected parallel to axle of the wheel, then the $Y$-axis will be parallel to the movement. The emitter width $W$ (the length of the side along the $X$-axis) should be equivalent to the wheel thickness $T$, and the height $H$ should correspond to the length $L$ of non-repeating part of the tread pattern. If there is no tread, the emitter height can be set arbitrarily within 5-10% of the wheel circumference. To solve the task of modeling tracks, we assume that sizes, position and orientation of each newly created particle is equivalent to the sizes, position and orientation of the emitter at the time of its generation.

In this work, particle systems are implemented based on two modern tools for multi-core GPUs. Creation, processing, storage and destruction of elements is carried out through the CUDA architecture with using compute cores of graphic processor and video memory resources. Particle visualization process, including geometric model creation, lighting and texturing, is performed on a graphics pipeline programmed by means of own vertex, geometry and fragment shaders.

Classic particle systems from 3D modeling software usually have a specified frequency of element generation, which can be set by scene designer or is calculated as the ratio between the total number of particles and maximum particle lifetime. This paper proposes own type of the system in which creation and display of each next particle is done only when the control signal is supplied. Let the total number of elements in such system equal to $n$, and each of them has its own integer timer $C$. A particle is inactive and not displayed, if $C \leq 0$. When initializing the system, an element with number $i \in [0, n\text{-}1]$ is assigned the value $C_i = -i$. At the time of receipt of the next control signal, all timers are increased by 1. After this, a particle with the value of $C = 1$ passes the stage of generation, including the primary computing of all its characteristics in accordance with current position of the emitter and chosen settings of the system. An element, whose timer is equal to $n$, is destroyed and then created again. The timer itself is reset to 1. Thus, the receipt of $n$-1 control signals limits the maximum existence period of each particle. In addition, each particle stores its actual lifetime in seconds. This is important for implementation of such system parameters that determine the laws of changing the characteristics of particles over time. For example, the speed of attenuation (changing transparency over time) provides the ability to simulate drying traces.

# 3. Obtaining contact characteristics of wheel with surface

Proposed approach of wheel track simulation includes two main stages. At first, it is necessary to find out whether wheel model is currently in contact with any solid surface of virtual environment and, if such contact takes place, to obtain its characteristics, which will be mentioned below in this section. It is advisable to implement the stage in dynamics system of modeling complex, since this system is responsible for interaction between virtual objects. The second stage consists of generating and rendering a textured particle at the found contact area by means of the visualization system. It will be discussed in detail in Section 4.

In this paper, the fact of contact between wheel model and object surface is established on the basis of collision detection using non-displayed bounding volumes (boxes, spheres, etc.). Note that only objects that are rigid bodies are considered. For each virtual object, a set of such volumes is created, the outer boundary of which approximately covers its entire surface. Only one bounding cylinder surrounds the wheel so that their axes coincide. When the wheel interacts with object surface, dynamics system at any time determines a set of points for checking collisions, which are located on both the outer and inner rims [12]. Among these points, we choose the point $P$ that corresponds to a deeper penetration of the wheel's bounding cylinder into one of the object's bounding volume and shift it along the cylinder axis to the middle cross section plane (Figure 2). The displacement is conveniently performed in the local coordinate system OCS of the cylinder, which is usually located at one of the bases, and the $Z$-axis is directed along its axis. Knowing coordinates $P_{wcs}$ of the point $P$ in the world coordinate system WCS, its coordinates $P_{ocs}$ in the OCS can be obtained via multiplying by an inverse of model matrix $M$ of the cylinder: $P_{ocs} = M^{-1} \cdot P_{wcs}$. To obtain the desired position of displaced point $P'$ in the WCS, it is necessary to change the $z$-coordinate $P_{ocs,z}$ by half the thickness $T$ of the wheel and perform multiplication by the matrix $M$: $P'_{ocs}$ =

($P_{ocs,x}$, $P_{ocs,y}$, $P_{ocs,z} \pm T/2$), $P' = M \cdot P'_{ocs}$. The sign of the change depends on which wheel rim initial point $P$ lies on. The resulting coordinates $P'$ will be used as the location point for the emitter of particle system that generates wheel track. In other words, the origin of the emitter's local coordinate system ECS will move to this point at the considered time moment.

To set the emitter orientation in accordance with the slope of the surface, it is also necessary to have information about normal vector $\boldsymbol{N}$ to it at the point $P'$. To simplify calculations, we use as $\boldsymbol{N}$ a normal to that part of the object's bounding volume, which bounding cylinder of the wheel crosses first (Figure 2). Next, we rotate the emitter so that the $Z$-axis of its coordinate system ECS coincides with $\boldsymbol{N}$.

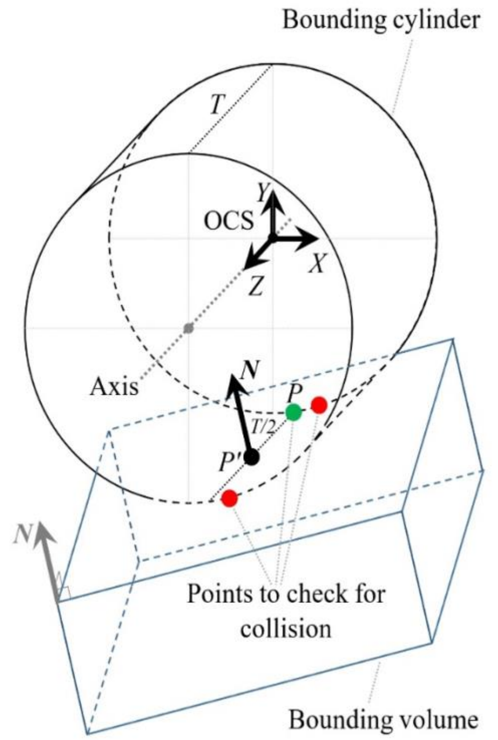


Fig. 2. Finding a point to place the emitter and normal for its orientation.

Based on the obtained data, dynamics system generates and sends a signal for visualization system about the presence of contact between the wheel and surface at current time. In addition, parameters of new position and orientation of the emitter are sent, as well as current angle φ of the wheel rotation around its axle. The latter is equal to zero at the moment of loading a virtual scene, increases or decreases when technical vehicle model moves forward or backward, respectively. If there is no collision of the wheel's bounding cylinder with bounding volume of any surface (a situation where the wheel is hanging in the air), such signal is not generated, and no data is sent.

Described method allows drawing traces always when there is contact of the wheel with any solid surface. This paper also proposes a solution for generating tracks only after interaction with a polluting environment. The essence of the approach is as follows. A material whose name contains special symbolic flag is applied to simulated surface area with a pollutant. When a collision is detected, dynamics system analyzes the presence of such flag for the surface material under the wheel. If there is no the flag and there was not before, the signal about the wheel contact with object is not sent to visualization system, and therefore the track is not drawn. As soon as the wheel is in polluting area, this fact is detected by dynamics system, and it begins to wait for the event when the wheel leaves this area. Negative edge of a signal about the symbolic flag presence indicates such event. Then user-configurable timer starts, during which the signal about a contact of the wheel with object is sent to visualization system. If the wheel again finds itself on polluted surface when time is counted, the timer is reset to its original value and does not work until the wheel leaves it.

## 4. Track element creation and visualization

As mentioned in the previous section, the second stage of wheel track simulation includes generation and rendering of textured particle (element of the track) at the contact area of virtual models of the wheel and solid object. Visualization system solves this task, which receives from dynamics system the signal about the presence of contact, position and orientation of the emitter of particle system associated with the wheel, as well as current rotation angle φ of the wheel.

To generate the next element of wheel track, it is necessary to form and send control signal to the appropriate particle system. Because all elements must lie on virtual surface one after another with no overlaps (except self-intersections of wheel path), forming a visible line of wheel path by means of the tread pattern, the signal about the presence of contact obtained from the dynamics system is necessary but not sufficient condition for creation of new particle. The second factor is a change Δφ in the wheel rotation angle φ that has occurred since the last generation of the track element. To create the next element, absolute value of Δφ in radians must reach the value $\alpha = 2\pi \frac{L}{2\pi R} = \frac{H}{R}$, where $R$ is the wheel radius, $H$ is the height of particle system's emitter, $L$ is the length of non-repeating part of the tread pattern (Figure 3). Thus, visualization system generates control signal in case of simultaneous observance of the following conditions: there is contact of the wheel with some object's surface, and $\Delta\varphi \geq \alpha$.

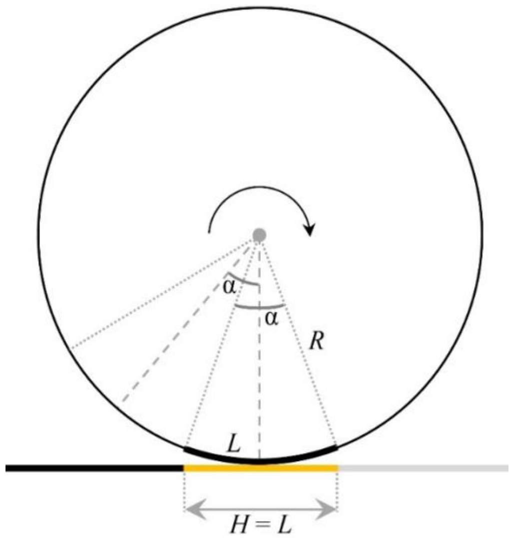


Fig. 3. Wheel rotation angle for generating new element of the track.

During the synthesis process of each frame of virtual scene, particle system associated with the wheel computes and stores the necessary parameters of its elements, including the lifetime of active particles. When control signal is given, values of the counters $C$ of all elements are also increased by one that leads to creation of one new element and destruction of one existing element, as was described in detail in Section 2. These actions are implemented with use of the CUDA parallel computing architecture. In this case, each particle is represented in the form of coordinates of central point $P'$ and three Euler angles, which determine its position and orientation in the WCS and are computed from the emitter's model matrix when generating considered particle. The aggregate of the center point, rotation angles, and lifetime forms data structure needed to render the particle. An array of such structures for all system elements is transferred from the CUDA context to vertex shader using technology of vertex buffer object (VBO).

Creation of geometric models for particles is performed at the stage of geometry shader. For this, the shader uses the Euler angles to restore rotation matrix of the emitter at the time of particle generation and obtains from it coordinates $\boldsymbol{X}_E$, $\boldsymbol{Y}_E$ of the $\boldsymbol{X}$, $\boldsymbol{Y}$ basis vectors of the ECS in the world coordinate system. Next, based on the emitter sizes $W$, $H$, positions of triangle strip's four vertices $V_0$, $V_1$, $V_2$, $V_3$ are computed, for which texture coordinates $s_i$, $t_i$, $i \in [0, 3]$ are set in such a way that the top of texture map assigned to the particle is located in the forward direction of vehicle movement (Figure 4):

$$V_0 = P' + 0.5(W \cdot \boldsymbol{X}_E - H \cdot \boldsymbol{Y}_E),\ s_0 = 1.0,\ t_0 = 0.0;$$
$$V_1 = V_0 + H \cdot \boldsymbol{Y}_E,\ s_1 = 1.0,\ t_1 = 1.0;$$
$$V_2 = V_0 - W \cdot \boldsymbol{X}_E,\ s_2 = 0.0,\ t_2 = 0.0;$$
$$V_3 = V_1 - W \cdot \boldsymbol{X}_E,\ s_3 = 0.0,\ t_3 = 1.0.$$

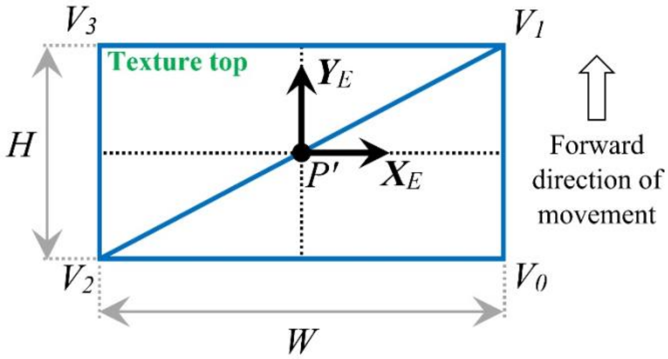


Fig. 4. Creation of particle geometry.

Depending on the lifetime, fragment shader may not render some particles at all, or change their rendering parameters, such as opacity value.

# 5. Results

Methods and approaches for simulation and visualization of wheel tracks from technical vehicles proposed in this paper were implemented in virtual environment system VirSim [13] developed at the SRISA RAS. Testing of developed solutions was carried out using a scene of the virtual polygon, which includes about six hundred thousand textured triangles. For this, three-dimensional model of wheeled robot was placed in it and also a puddle of spilled liquid was implemented (Figure 5).

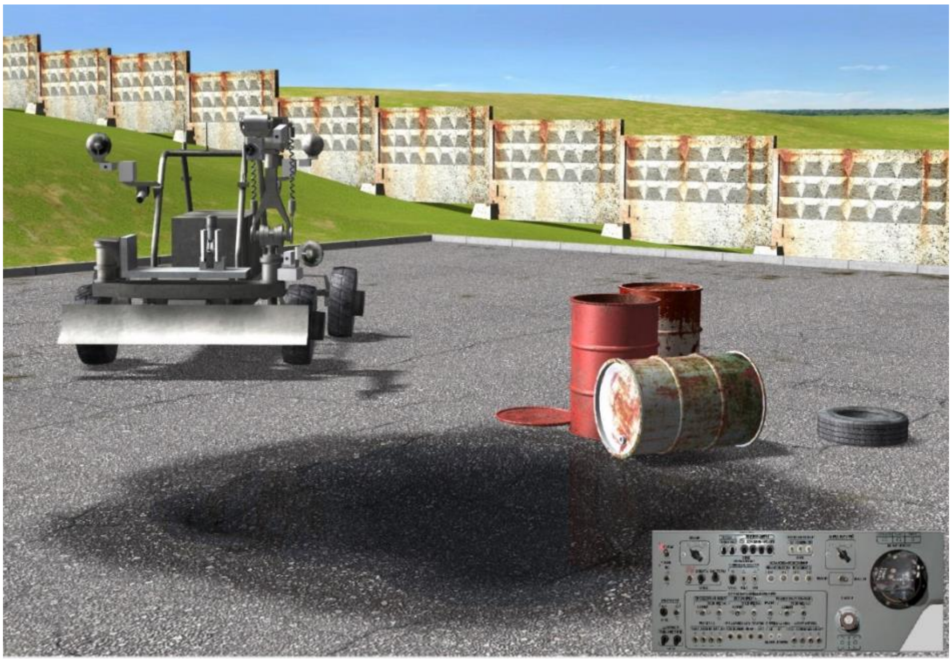

Fig. 5. Scene of the virtual polygon with wheeled robot and spilled liquid.

User can control the model by means of computer mouse, acting on elements of virtual control panel [14]. This panel is shown in the lower right corner of Figure 5. Since the tread pattern for left and right wheels of used robot is different, two opacity maps (textures) are used to simulate traces from them. Figure 6 illustrates these textures. Black color on maps corresponds to full transparency, and gray color corresponds to shading with opacity factor of 0.43. In addition, we use black as the diffuse color of wheel tracks.

Fig. 6. Left and right wheels of 3D robot model and corresponding opacity maps for elements of tracks.

When creating the puddle, approaches based on multitexturing of surfaces, reflection and opacity maps were used. A material of the polygon's surface area located directly under the liquid layer is marked with special symbolic flag. The flag notifies dynamics and visualization systems to start generating and displaying track from wheel, interacting with this area, after the wheel leaves its boundaries.

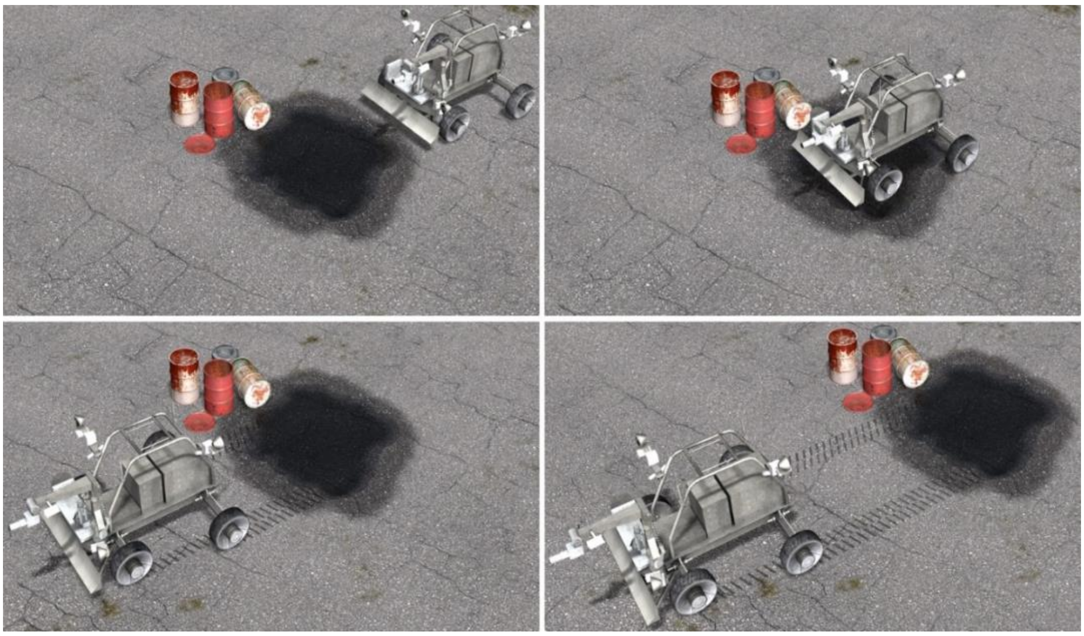

Fig. 7. Wheel tracks from robot that drove through spilled liquid.

Figure 7 shows how virtual model of the robot drives through the spilled liquid, followed by the formation of wet tracks on the asphalt. To evaluate the performance of proposed solutions, we carried out a number of frame rate measurements for the VirSim complex using both one and several instances of the robot model in the virtual polygon scene. Three different scenarios were considered. The first one is without wheel tracks to get a reference performance for the scene in question. The second one is standard. In this case, wheel track can contain a maximum of 10k elements, what is sufficient for most scenes. The third is high overloaded scenario in which 500k particles are allocated for one wheel. Tests were performed using GeForce RTX 2080 Ti graphics card. Table 1 presents the results obtained.

Table 1. VirSim performance measurements for the virtual polygon scene.

| **Number of robots** | **Particle number per wheel** | **Total number of particles for wheel tracks** | **Performance** (dynamics+visualization), *ms* |
|---|---|---|---|
| 1 | 0 | 0 | 15 |
| | 10k | 40k | 18 |
| | 500k | 2000k | 21 |
| 4 | 0 | 0 | 19 |
| | 10k | 160k | 27 |
| | 500k | 8000k | 38 |

Approbation of proposed methods and approaches in the VirSim software complex showed that they are effective for simulation and visualization of wheel tracks from technical vehicles in virtual environment systems, training complexes and other applications.

## 6. Conclusions

This paper presents original methods for implementing wheel tracks from technical vehicle models on virtual solid surfaces. The advantage of proposed solutions is use of particle systems with controlled generation of elements, and parallel computing on GPU, that allows real-time making long traces without changing the original geometry and retexturing scene objects. Results obtained in the paper can be used in software development for training complexes and virtual environment systems.

## 7. Acknowledgements

The publication is made within the state task of Federal State Institution “Scientific Research Institute for System Analysis of the Russian Academy of Sciences” on “Carrying out basic scientific researches (47 GP)” on topic No. FNEF-2022-0012 “Virtual environment systems: technologies, methods and algorithms of mathematical modeling and visualization. 0580-2022-0012”.

## References

1. Bruguera M.B., Ilk V., Ruber S., Ewald R. Use of virtual reality for astronaut training in future space missions – spacecraft piloting for the Lunar Orbital Platform – Gateway (LOP-G) // 70th International Astronautics Congress, Washington D.C., 2019.
2. Maltsev A.V. Computer simulation of video surveillance complexes in virtual environment systems // Scientific Visualization, 2022, Vol. 14, No. 2, pp. 88-97.
3. Mikhaylyuk M.V., Timokhin P.Yu. Memory-effective methods and algorithms of shader visualization of digital core material model // Scientific Visualization, 2019, Vol. 11, No. 5, pp. 1-11.
4. VR trenazher. Virtualnaia realnost v obuchenii. Neftianaia promyshlennost [VR simulator. Virtual reality in learning. Oil industry]. https://www.youtube.com/watch?v=keXlfMKyxsI. Accessed 27 February 2023. [in Russian]
5. Garcia A.D., Schlueter J., Paddock E. Training astronauts using hardware-in-the-loop simulations and virtual reality // AIAA SciTech Forum, Orlando, FL, 2020.

6. Maltsev A.V., Strashnov E.V., Mikhaylyuk M.V. Methods and technologies of cosmonaut rescue simulation in virtual environment systems // Scientific Visualization, 2021, Vol. 13, No. 4, pp. 52-65.
7. Chen G., Zhang J., Xu X., Yin Y. Real-Time Visualization of Tire Tracks in Dynamic Terrain with LOD // Technologies for E-Learning and Digital Entertainment. Lecture Notes in Computer Science, 2007, vol. 4469, pp. 655-666.
8. Wang D., Zhang Y., Tian P., Yan N. Real-Time GPU-Based Visualization of Tile Tracks in Dynamic Terrain // 2009 International Conference on Computational Intelligence and Software Engineering, Wuhan, China, 2009.
9. Xia Y., Chen Y., Wang D. Real-Time LOD Rendering of Tire Tracks in Dynamic Terrain // 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), Xiamen, China, 2019, pp. 206-209.
10. Creating Tire Tracks with Particles in Blender 2.92. https://www.youtube.com/watch?v=6EEvSMXzSv8. Accessed 27 February 2023.
11. Blender. Documentation. https://www.blender.org/get-involved/documentation/. Accessed 27 February 2023.
12. Strashnov E.V., Torgashev M.A., Maltsev A.V. Opredelenie kollizii approksimiruiushchikh parallelepipedov i tsilindrov [Collision detection of bounding boxes and cylinders] // Trudy NIISI RAN, Vol. 10, No. 5-6, 2020, pp. 139-146 [in Russian].
13. Mikhaylyuk M.V., Maltsev A.V., Timokhin P.Ju., Strashnov E.V., Krjuchkov B.I., Usov V.M. Sistema virtual'nogo okruzhenija VirSim dlja imitacionno-trenazhernyh kompleksov podgotovki kosmonavtov [The VirSim virtual environment system for the simulation complexes of cosmonaut training] // Pilotiruemye polety v kosmos, Vol. 4, No. 37, 2020, pp. 72-95 [in Russian].
14. Mikhaylyuk M.V. Dvumernye virtualnye pulty upravleniia v trenazhernykh kompleksakh [2D Virtual Control Panels in Simulators] // Programmnaia inzheneriia, No. 5, 2014, pp. 20-25 [in Russian].