Skeletal Animation with Movement Disorders in Blender and OpenGL

Abdulrahman Emad Abdulfattah
Systems and biomedical Engineering
Cairo University
abduelrahmanemad@gmail.com

Omar Tarek Mahmoud
Systems and biomedical Engineering
Cairo University
omar.elsayed102@eng-st.cu.edu.eg

Mourad Magdy
Systems and biomedical Engineering
Cairo University
murad.ibrahim02@eng-std.cu.edu.eg

Ziad Ahmed Meligy
Systems and biomedical Engineering
Cairo University
line 5: email address or ORCID

Yousef Ashraf
Systems and biomedical Engineering
Cairo University
line 5: email address or ORCID

Abstract— **This study explores making realistic character movements in Blender and OpenGL, simulating disorders like Parkinson's. We use Blender for animation and OpenGL for quick visualization, creating accurate skeletal models and movements. We compare our virtual movements with real patient data for validation. Our work benefits computer graphics and can help in medical education by making it easier for healthcare pros to understand movement disorders. Blender and OpenGL together offer a handy platform for practical simulations, useful in entertainment and education.**

Keywords: 3D modeling; Blender; OpenGL; Movement Disorders; Movement Disorders simulations Animation.

I. Introduction

In the realm of computer graphics, the integration of skeletal animation techniques with the simulation of movement disorders holds great promise for creating more realistic and meaningful virtual environments. Skeletal animation has become a cornerstone for lifelike character movements in various applications, ranging from entertainment to medical education. This study endeavours to push the boundaries of this technology by infusing it with the intricacies of movement disorders, such as Parkinson's disease and muscular dystrophy. By employing the powerful combination of Blender for character design and animation and OpenGL for real-time rendering, we aim to not only enhance the visual realism of virtual characters but also contribute to the understanding and recognition of movement disorders in a simulated environment. This introduction outlines our motivation, the significance of the research, and the key technologies at the core of our exploration.

II. Literiture Review

The synergy between Blender and OpenGL in simulating movement disorders offers an innovative approach to bridge the gap between computer graphics and medical science. Researchers leverage the flexibility of Blender's animation tools to model intricate skeletal movements, while OpenGL's real-time rendering capabilities enable dynamic visualizations. This interdisciplinary integration not only aids in the accurate representation of movement disorders but also fosters collaboration between computer graphics experts and medical professionals. The resulting simulations can be valuable for educational purposes, enabling healthcare practitioners to better comprehend the nuances of various movement disorders and potentially contributing to the development of more effective therapeutic interventions. The convergence of Blender and OpenGL in this context showcases the potential of computer graphics in advancing medical visualization and understanding movement disorders.

Existing research in this domain has explored various methodologies to integrate skeletal animation with movement disorders in Blender and OpenGL. Techniques such as skinned instancing and vertex skinning have been employed to optimize the rendering process, enhancing computational efficiency and reducing the number of draw calls required for rendering agents. While the primary focus lies on skeletal animation for crowds, the literature also introduces methods to render characters at lower levels of detail (LOD), incorporating image-based approaches and impostors to simplify the rendering process. These advancements not only contribute to the field of computer graphics but also hold potential for practical applications in medical education, diagnosis, and treatment planning.

III. METHODOLOGY

The process of animating movement disorders in Blender consists of three main steps: (i) building the character and the skeletal structure; (ii) animating normal movements and disorders; (iii) verification and validation. The process of integrating the results with OpenGL consists of four main steps: (i) exporting from Blender; (ii) setting up OpenGL environment; (iii) rendering the 3D model; (iv) user interactions and navigation.

Building the character and the skeletal structure: Utilizing Blender’s modeling tools to create a 3D representation of the human body, ensuring anatomical accuracy and attention to detail. Then construct the Skeletal Framework with the armature system of Blender defining bones, joints, and connections to establish a robust foundation for animation. Finally Rigging the character by assigning weights to different parts of the mesh to correspond with the bones in the skeletal structure.

Animating Normal Movements and Disorders: Defining the key poses for each movement. This involves identifying critical points in the animation sequence that convey the essence of the movement. Then with Blender’s keyframing capabilities to set keyframes for the established key poses. This defines the starting and ending points of each movement, allowing Blender to automatically generate intermediate frames or in other words interpolating the frames. After setting the keyframes, Refine the Animation Curves to achieve smooth and realistic movement. Finally test the animation to identify any discrepancies or issues.

Verification and Validation: Compare the animated movements with reference material, such as real-life observations or videos to ensure accuracy and realism. If possible, seek input from experts or individuals with knowledge in anatomy and movement disorders, finally gather feedback from end-user target audience.

Exporting from Blender: Choose an appropriate file format for exporting the 3D model and animation from Blender. OpenGL compatible with many types such as .fbx, .obj or .glTF

Setting up OpenGL Environment: Implement the necessary code to initialize an OpenGL context. Set up rendering parameters, viewport, and any required shaders. If possible, integrate any external libraries that assist in loading 3D models and animations.

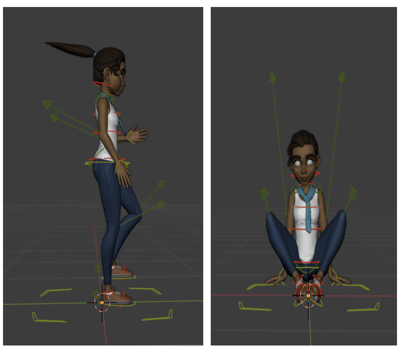
Rendering the 3D Model: develop code to load the 3D model into the OpenGL environment. Update the transformation matrices based on the exported animation data.

User Interactions and Navigation: Incorporate user input handling to allow interaction with the 3D scene.

Verification and Validation: After integrating the results into the OpenGL environment, perform a final round of verification and validation. Ensure that the skeletal animation accurately translates from Blender to OpenGL. Compare the visual representation in OpenGL with the animations generated in Blender, and validate against reference materials. Seek input from experts or end-users to confirm the accuracy and realism of the integrated results.

User Testing and Feedback: Conduct user testing with the target audience to gather feedback on the interactive experience. Pay attention to user interactions, navigation preferences, and overall satisfaction with the simulated movement disorders. Iterate on the OpenGL implementation based on user feedback, making necessary adjustments to enhance usability and realism.

Documentation and Reporting:Document the entire process, including details of character modeling, skeletal structure creation, animation development in Blender, export steps, OpenGL integration, and user interaction implementation. Provide comprehensive documentation on code structures, algorithms used, and any specific configurations.



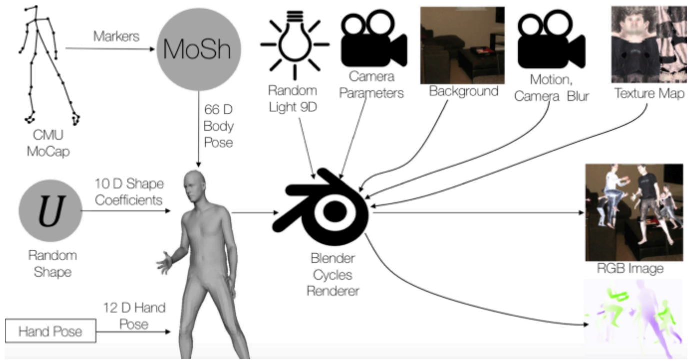


Figure 2. Process map for the rendering simulation of a 3D human body model in blender

IV. EXPERIMENTS AND RESULTS

A. OpenGL and Importing the model:

In our attempts to integrate Blender output into our OpenGL environment, we explored the use of third-party libraries, with a particular focus on the Assimp (Open Asset Import Library). However, the implementation posed several challenges. First and foremost, the library's dependency on multiple external libraries created hurdles during the installation process. Though eventually overcoming these issues, we encountered additional challenges. The Assimp library necessitated the implementation of custom shaders, adding a layer of complexity to the integration process. Furthermore, the requirement for these custom shaders brought about unexpected complications, including compatibility issues and difficulties in synchronizing animations. Ultimately, we opted to create our own parser. We developed two parsers – one for the .obj extension and another for the .mtl extension. The results from these parsers proved satisfactory for proceeding with our project. Achieving these results necessitated a comprehensive study of the structure of each extension. For instance, within the .obj extension, distinct symbols represent vertices (v), textures (vt), normals (vn), and faces (f), with faces being composed of vertex indices, texture coordinate indices, and normal indices. Figure 1 and Figure 2 showcase the successful outcomes obtained using our custom parser, affirming its efficacy with both .obj and .mtl extensions. During our analysis, we ensured the inclusion of all relevant keys, accounting for the nuances of each format. In the context of the .mtl file, the parser meticulously addressed key properties such as ambient color (Ka), diffuse color (Kd), specular color (Ks), shininess (Ns), and texture mapping (map_Kd). This thorough consideration ensured a comprehensive and accurate parsing process, leaving no key properties overlooked.

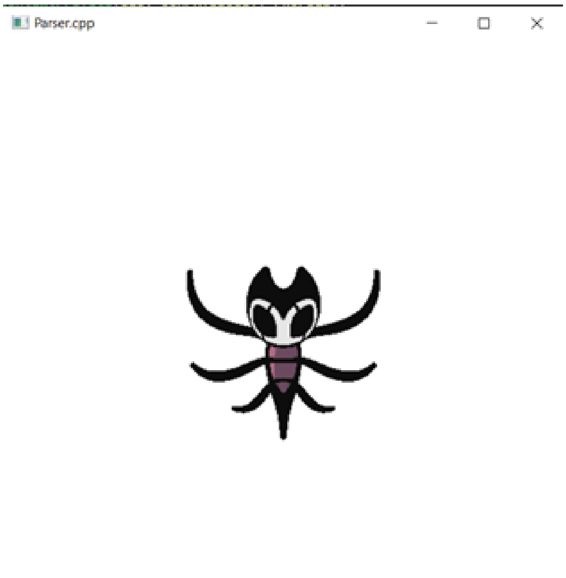


Figure 1 Result from custom parser` with texture.

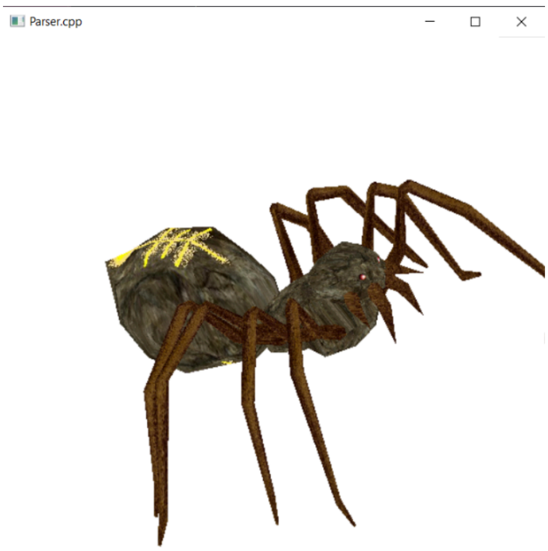


Figure 2 Another Result from custom parser with texture.

Here are the results in figure 3 and figure 4 without the texture and showing different types of connections between vertices. In figure 3 the vertices are connected with lines.

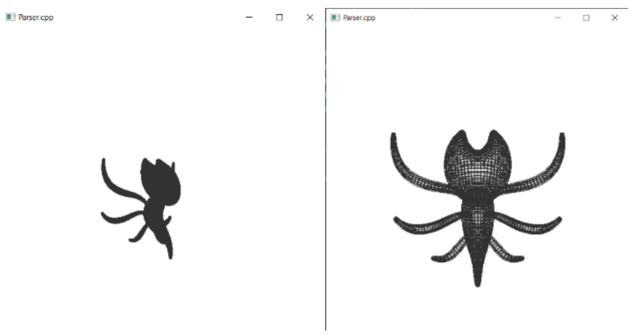


Figure 3 Result from the parser without the texture.

In figure 4 the vertices are connected with triangles.

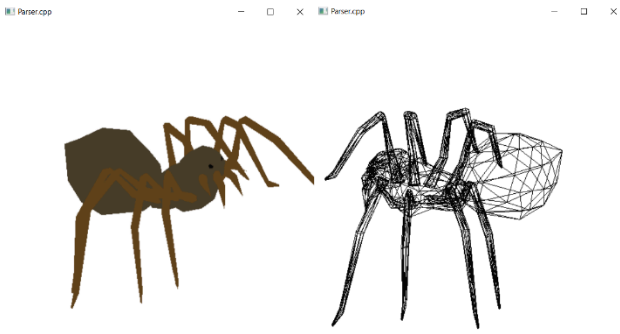


Figure 4 Result from the parser without texture.

However, as we progressed, we discovered that these extensions did not encompass animation data. Consequently, we chose to exclusively continue our work within Blender, recognizing the necessity of a more tailored approach when animation intricacies were paramount. This decision highlighted the importance of thoroughly evaluating the capabilities and limitations of external tools, reinforcing the need for a customized solution to meet our specific requirements.

B. Blender and Visualizing skeletal human body and simulating animations of body

Creation of Realistic Human Models and Dynamic Skeletal Structures in Blender:

Our exploration commenced with the meticulous creation of a human body model in Blender, enriched with a sophisticated skeletal structure. Utilizing Blender's powerful Rigify tool, we attached a meta rig to the model, providing extensive freedom of movement for various bones. This meta rig, intricately bound to the model, facilitated the animation of a wide array of movements, including those of the hands, legs, head, and more, enabling us to simulate the complexities of the human body.

Starting with the model, we initiated our process by employing a highly detailed meshed model designed to closely resemble the anatomical features of the human body. The model boasts a substantial number of vertices, ensuring precision in capturing the nuances of human anatomy. Careful attention was given to factors such as limb proportions, joint articulations, and overall body morphology, laying the foundation for a realistic representation.

To enhance the model's versatility and range of motion, we seamlessly integrated Blender's Rigify tool. Rigify provided an efficient and adaptable solution for the creation of a dynamic skeletal structure. The meta rig, intricately connected to the model, allowed for the manipulation of various bones, granting extensive freedom to simulate a myriad of movements realistically. This approach not only expedited the animation process but also ensured a lifelike quality in the portrayal of diverse bodily actions.

The newfound model exhibits a level of detail that goes beyond the standard metarig, particularly in its representation of facial and body muscles. With a meticulously crafted mesh and an increased number of vertices, this skeletal model enables an unprecedented level of granularity in capturing the subtleties of muscle movements. The high resolution facilitates a nuanced portrayal of facial expressions, intricate hand gestures, and intricate body movements that contribute to a more lifelike and expressive simulation.

Unlike conventional rigify metarigs, this advanced model provides an extensive array of controls, allowing for the manipulation of individual muscles. This level of granularity empowers us to simulate a wide variety of facial expressions, ranging from subtle smiles to intense frowns, as well as intricate body expressions that encompass the complexities of natural human motion. The model's adaptability allows us to tailor movements to specific scenarios, capturing the essence of diverse emotions and physical actions with unparalleled fidelity.

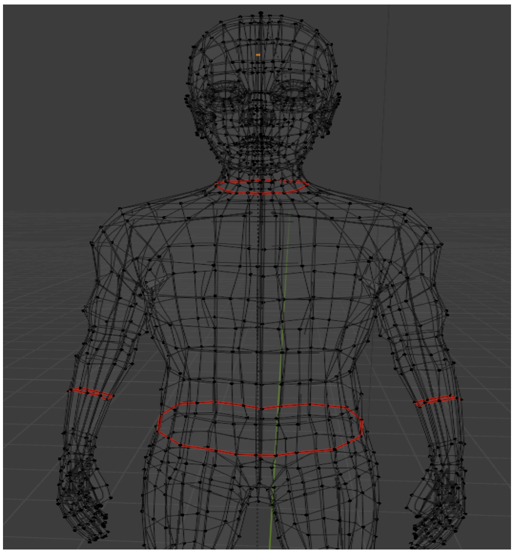


Figure 5. First result of the wireframe of human model

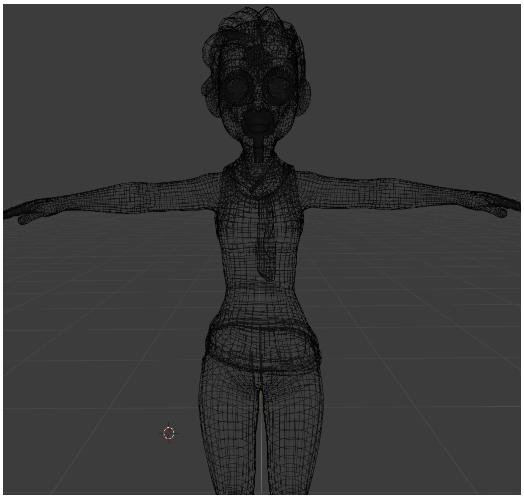


Figure 6. Wireframe of a more advanced model with substantial more vertices resulting in a satisfying high quality.

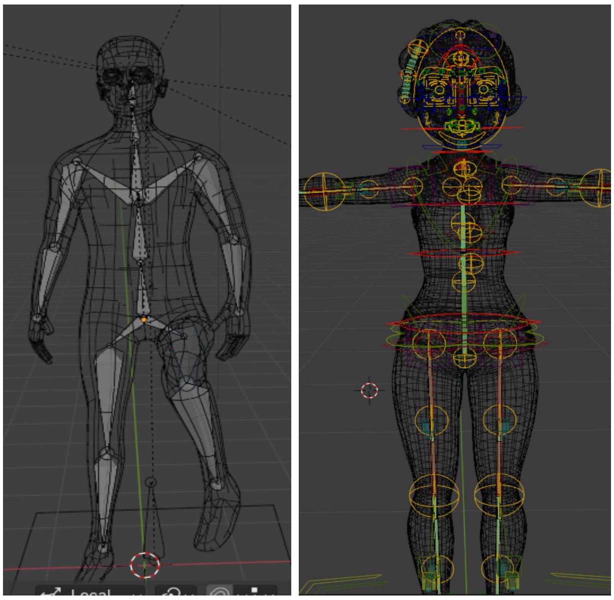


Fig 7. This figure illustrates the extensive array of controls the enhanced model (on the right). Each point colored point in the image manipulates an individual muscle which makes it more superior than our base model using rigify's meta-rig which has little range of freedoms for the model to move.

After snapping the an appropriate rig that gives us a plethora of options to manipulate any pose we wish for. We utilized Blender's keyframe animation techniques to meticulously craft sequences that encapsulated the fluidity and natural transitions between different poses. Timing graphs and bode sheets were employed to fine-tune the temporal aspects of the animations, ensuring a seamless and lifelike representation of the chosen actions by using linear interpolation.

These methods resulted in seamless life-like simulation of a range of activities of humans. e.g. walking, jumping, sitting, running etc

Fig 8. shows the model sitting on the ground and running as a collection of animations of everyday movements.

We have looked at scientific papers over popular disorders that affect movement of human's greatly handicapping them. It is very important to study these disorders further. as a result, we took our simulations a step further, we delved into the animation of two specific movement disorders – X-Linked Dystonia Parkinsonism and Medjesyndrom Parkinson. Leveraging Blender's keyframe animation features, we meticulously animated the manifestations of these disorders. Utilizing timing graphs and bode sheets, we ensured a thorough representation of the associated symptoms, capturing the intricacies of these neurological conditions

Fig 9. shows one of the disorders (X-Linked Dystonia Parkinsonism) as an image. It is a pose handmade captured from the many frames in the frames making the animation.

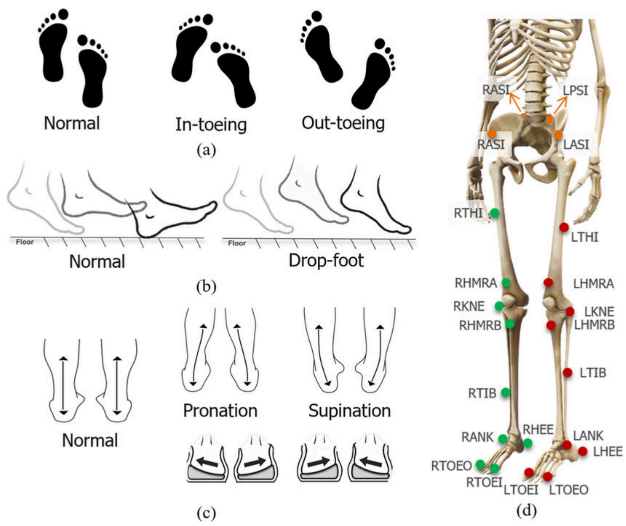


Fig 10. This figure is a visualization for various of abnormal gait patterns that was used to understand and diagnose patients with specific disorders like parkinson's

The process of animating a disorder was very tough as it needed a deep understanding of specific joint movements. This resulted in enhancing the knowledge of culprit joints and neurological disorders hoping for a cure or a solution as we demonstrate these disorders with higher resolution of understanding through visualization to researchers and doctors.

The success in creating a dynamic Blender model, coupled with thorough animations of movement disorders, holds significant implications for computer graphics and medical education. The versatility of the Rigify meta rig and the precision of Blender's animation tools provide a valuable resource for healthcare professionals and educators. Our experiments showcased the potential for enhancing medical education through immersive simulations of specific movement disorders.

V. Conclusion

The synergy of our innovative approach in creating realistic human models and simulating movement disorders presents multifaceted advantages for both computer graphics and medical education:

The detailed and accurate representation of human movements in our simulations provides a valuable resource for computer graphics professionals, allowing them to craft more lifelike animations. In the realm of medical education, healthcare professionals gain enhanced visualization and understanding of movement disorders, fostering a deeper comprehension of their clinical presentations.

The animated simulations of movement disorders serve as a valuable tool for healthcare professionals in honing their diagnostic skills. By closely observing and analyzing the nuanced symptoms presented in our simulations, medical practitioners can refine their ability to recognize and differentiate various movement disorders, ultimately improving patient care and treatment strategies.

Our work opens up new avenues for research and development in both computer graphics and medical fields. Computer graphics researchers can leverage our detailed models and animations to explore innovative techniques, while medical researchers can utilize our simulations for studying the biomechanics of specific disorders, potentially leading to advancements in therapeutic interventions.

In conclusion, our experiments involved the creation of a realistic Blender human model, equipped with a dynamic Rigify meta rig, and its integration with OpenGL for visualization. The animations of specific movement disorders demonstrated the versatility and precision of Blender's animation features, with potential applications ranging from medical education to broader entertainment and educational contexts.

VI Member contributions

Mourad Magdy:-

running animation,Jumping animation,literature review

Omar Tarek:-

sitting animation, medjesyndrom parkison, blender methodology

Youssef Ashraf&abdelrahman emad:-

built our own OBJ and MTL parser, blender enviroment and light,experiments and results in opengl

Ziad Meligy:-

Walking animation and parkinson disorder,experiments and results in blender

VI. References

[1]H. Voss, “The representation of movement disorders in fictional literature: Table 1,” *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 83, no. 10, pp. 994–999, Jun. 2012, doi: https://doi.org/10.1136/jnnp-2012-302716.

[1]C. Paduraru and M. Paduraru, “Techniques for Skeletal-Based Animation in Massive Crowd Simulations,” *Computers*, vol. 11, no. 2, p. 21, Feb. 2022, doi: https://doi.org/10.3390/computers11020021.