UDC 004.94:004.92
DOI: 10.25140/2411-5363-2020-3(21)-220-228

Erik Prada, Adam Kolář

POSSIBILITIES OF CONVERT CAD MODELS FOR REAL TIME RENDERING SOFTWARE

Urgency of the research. *The proposed methods can save valuable time and cut down the cost on hardware in some instances and make it easier for a non-technical group of professionals to design and renderer products.*

Target setting. *To use free and paid products for use in modern businesses and schools to optimize the design and research phase and also create environments for training the qualified personnel in extreme conditions.*

Actual scientific researches and issues analysis*. These optimizations are done directly by developers of the real time simulation tools and some large companies but a lot of them are fractured to smaller businesses and need one direction to make it to the mainstream.*

Uninvestigated parts of general matters defining. *Conversion methodology CAD models for real time rendering.*

The research objective. *Make a uniform standard of procedures on how to leverage the power of real time rendering tools and find ways to use them in industrial practice.*

The statement of basic materials*. Possibilities of using the game engine Unreal Engine and the Epic Games addon DATASMITH for conversion CAD models for real time rendering software are presented in article*

Conclusions. *The game engines can be used with right tools to make it easier for artists and other non-technical professions to see results immediately. Also these tools can be leveraged to make simulations of various kinds with higher efficiency but the results can be less accurate. But with the development of better tools and better integration of CAD software these results can be improved and me and also other professionals believe that the future of the designing and testing products is real time.*

Keywords: *CAD model; DATASMITH; Unreal Engine; UV map; Virtual Reality.*
Fig.: 9. References: 11.

Introduction. In many industrial use cases there is a need for using different tools and applications to reach a certain goal in research and development of product or mechanical assemblies. In mechanical engineering these problems are common in industrial software. For example, if you create a CAD model in SOLIDWORKS and you need structural analysis, it might happen that particular software doesn't have the best tools for your specific use case. Then you as an engineer are forced to use other expensive software to make these analyses. This problem is encountered most commonly in the process when we need a photorealistic render of a product or we want to create a model from CAD software to use it in a game engine for real time rendering. These demands for this kind of rapid development will only rise in the near future. If we use the right tools, we can significantly cut time needed for development. Also artists now can create realistic graphics or promotional renders with photorealistic fidelity by touching a button. There were many advances in this field of real-time rendering, but we would like to highlight a particular use case, which shows what these tools in the right conditions can achieve. The Michigan based Mechanical Simulation firm uses Unreal Engine and their proprietary software CarSim thanks to which they can simulate accurate conditions for driving such as asphalt type or road slope etc. Their software is splitted to two big softwares. The solver which solves the physical problems and Unreal Engine which creates the visualization. This firm shows that these kinds of simulations are possible and useful, for example as simulating autonomous driving and using that data to improve real product. The customer of VERTechs firm were able to create virtual city where they tested their autonomous vehicles. This is just one of the many possibilities which can be done by an Unreal Engine. This article will be exploring the possibility of solving the fundamental problem which is introduced by using a game engine [1-4].

Differences between CAD simulations and real time methods. CAD software has many advantages then real time methods, but the real time tools are quickly catching up. When we think about creating a product in CAD software, for example SOLIDWORKS, we usually think about this workflow. We design the model, then define variables we want to simulate. Then we tell the software to calculate the results. We wait for the results to be calculated and then reiterate if we see structural deformations or other design flaws, and do this over and over again. CAD software makes it easier in recent years but the next evolution can be the real time rendering. When we think about real time rendering, the first thing that we see is used in the video

game industry. And that is not a bad thing. These tools used in this particular industry are very powerful and have a lot of foundation already to be used in the professional spaces. A great example is a tool called ArchViz, in which we can transfer integrity from application that is used for building designs to a game engine and render the building in real time so that clients can walk inside a building, while this building is not even built. Now we apply this promise of high fidelity simulation to engineering. The workflow will be in the high level the same. We design the product, then we test and repeat this process. But the benefits start to show when we look closer to the parts of the process. After we have designed the product in a conventional CAD, we export it to a real-time renderer, which in our case is the Unreal Engine and in future or perfect case we have advanced tools in Unreal, thanks to which we do not need to further edit this CAD model. Then we specify what we want to start the test and we see real time results. For example, a plane crashing to the ground in real time and we can see the results of the deformation of the plane elements as soon as the plane hits the ground. Or we can pause the simulation and change for example the type of the engine in the middle of the simulation. These benefits of real time rendering can change our ways of how we approach research and development, and make these processes more straight forward [5].

The problem of the fragmentation of the software. When we worked on the project, which was focused on creating simulation software for a laic audience, we came across this problem. Being mechatronics engineers, we didn't want to learn a new software, which is specific for the game industry to model models for this use case. We are pretty familiar with CAD software, so that was a natural option. Many engineers had this same problem and EPIC games created solutions for this, but that is only a part of the solution. The other part is how we use these models. There are two ways, which will be discussed in the capitol below. But to outline the problem, we need to understand how these software works. CAD software has its own file format and the game engine uses their own standards. CAD software uses format specific for each platform, for example IGES, etc. The game software uses FBX. We needed a compatibility layer to convert these file formats and give them some other properties, which we also discuss below. The fundamental problem was the difference of file formats, but in many cases further modification of the basic design is required, but this is in the end worth the trade-offs [6, 11].

Available software and methods to convert CAD Models for real time rendering. There are many methods on how we can approach this problem. Some can produce usable results and some can produce results, that reach industrial standards. First, we need to choose the suitable render engine for our use case. If we look for commercial solutions, we have two biggest options and these are Unity and Unreal Engine. Our method was tested in Unreal Engine, because this engine has a necessary set of tools, which in our opinion are essential for this use case. If we want to render our CAD model in real time, we have two paths to export our model.

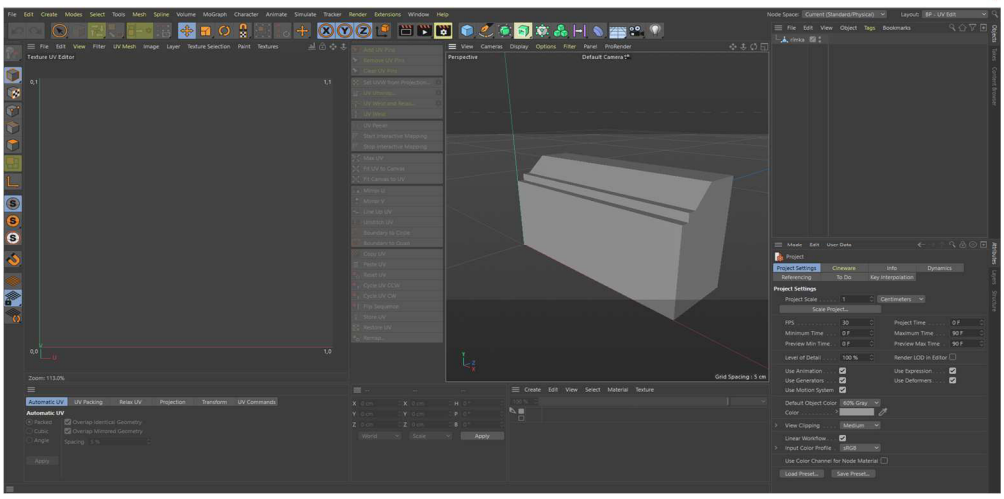

Fig. 1. STL model in Cinema 4D UV map view [6]

First path is to use the STL format for our model and then we import it to 3D editing software like Blender or Autodesk MAYA, in which we export our STL file as FBX file format. There are two significant problems with these methods. First one is UV mapping of the STL model and the second is file format itself. These problems only exist if we don't edit these models in Blender and Autodesk MAYA or Cinema 4D, and we only export them without fixing them. UV mapping is essentially telling the render engine how and on what scale the texture of the material should be applied to our model. The problem with that is that STL format is not exporting UV maps in SOLIDWORKS. You need to create it. For example, Cinema 4D can automatically create the correct UV map based on your model. Then if you export it as FBX file format and import it to the Unreal Engine and apply a material to it, it shows the material correctly, but it has some imperfections. We found that radial surfaces usually have the material blurred. Your results may vary, it all comes down to the geometry of your CAD model. The other problems are created directly from STL format and the way it handles the models. STL format is most notably used for 3D printers and the model is exported as layers, which creates all sorts of problems. Especially curved geometry is created from small triangular planes and if we want some degree of visual polishness, then we need a very large number of these planes, so the object is perfectly smooth to the human eye. That is a problem in UV maps. Because of this the UV map is far more complex than it needs to be and the editing of it becomes far more difficult. Also this creates visual inconsistencies that can be seen as blurred textures or stretched texture on the edges. To fix this problem we need to usually create a new UV map from scratch. This process is time consuming because we need not only to fix this inconstancy, but optimize the whole UV map for the large number of triangular planes. There are more methods to do this, but the best one from our results is to use the Unreal Engine plugin called DATASMITH. This method is created for the Unreal Engine model called Static mesh actor, which is commonly used in video games as FBX (filmbox) file format. FBX uses C++ and python so thanks to that this format is modular and can be used in open source editors such as Blender. This translation layer DATASMITH not only saves time, but makes models like arches or difficult geometry better, because it is created from polygons. That means game engines can render these models in high quality and thanks to next gen features like ray tracing, they can achieve high fidelity visuals and lighting. This add-on also removes the need for fixing UV maps by reducing these time consuming tasks from professional workflow. But we still may need to create UV maps, if we want to achieve professional results. Thanks to that, customers can ride a new car before it even hits the assembly line. The most notable problem is that relations between parts needs to be reapplied. That means if we for example export assembly of large 100 part assembly from SOLIDWORKS, then if we use the first method, we have the full object in the Unreal Engine. The relations between objects we need to fix and apply from the ground up. That said we can also import architecture data in the Unreal Engine and make archive simulations for clients. We found it best to apply material in the Unreal Engine, because thanks to Quixel Megascans you can create photorealistic scenes that look and feel real. If we add raytracing for lighting, then we have one of the best looking sceneries, which no CAD software can do. The best part is everything is real time rendered, so we can go to the scene and change static meshes and see the result right away. But this workflow can't completely eliminate the need for modeling software [8, 10, 11].

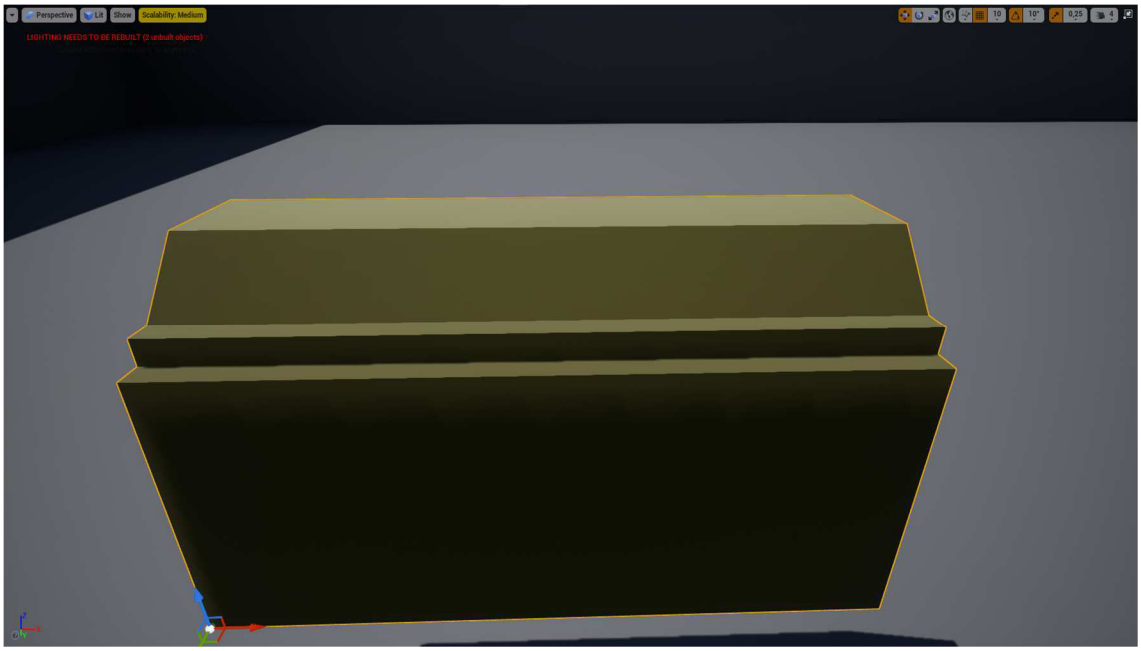

Fig. 2. STL model in Unreal Engine without UV map. Raw import [7]

We use this workflow to create levels for VR experiences and it saves time for our projects, because we create simple shapes in SOLIDWORKS and then in an Unreal Engine we create buildings. The benefit is that the DATASMITH keeps the dimensions in scale so, for example, 1 meter big in xyz cube in SOLIDWORKS is exactly similar and because of that the whole experience can be designed for example the scale of real building. These static meshes have advantage because then they can be vertex painted and the added layer of detail contributes to more photorealistic presentation.

Fig. 3. Scheme of software use in the particular order [8]

As we said, Quixel Megascans, which are free for Unreal Engine, also save significant time. These textures include information such as roughness, metallic and normal. Roughness is important because, as the name suggests, it changes meaning or shape, how rough a surface is, and how aggressively the light from that object is diffused. Normal map contributes to the physical shape of the model which can imitate, for example, a brick or stone surface. If correctly applied, these materials can increase the level of visual fidelity and make the final product more realistic. If we compare the end result between our method and the method of using DATASMITH, the results are pretty similar. The only difference is that by using this plugin we saved time and we had to do less work [9].

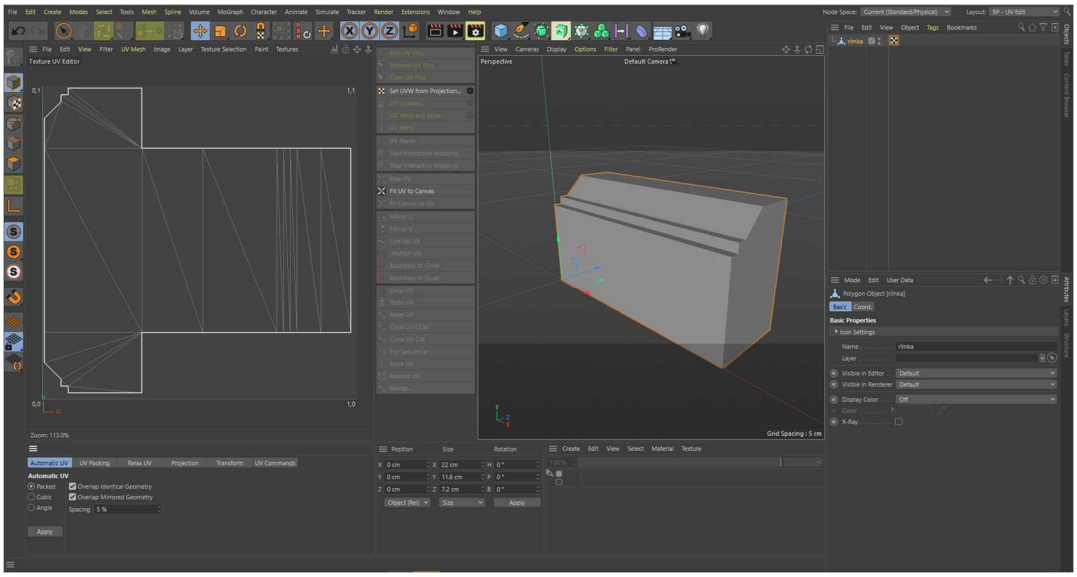

Fig. 4. STL model in Cinema 4D with uv map [9]

The difference of the STL and FBX file format. If we investigate the differences of these two formats, at the end we will see why the FBX format is more commonly used in game development. STL file format is used usually in stereolithography in CAD and has information only about the shape of the product. There is no information about the color or texture and other information like animations or audio add-ons. On the other hand, FBX file format, which is owned by AUTODESK, can store motion capture data, video data, animation data, audio data and usually is more commonly used in video games. Thanks to that, the FBX file format can be used not only for solid objects but for characters and other complex assets used in this type of application.

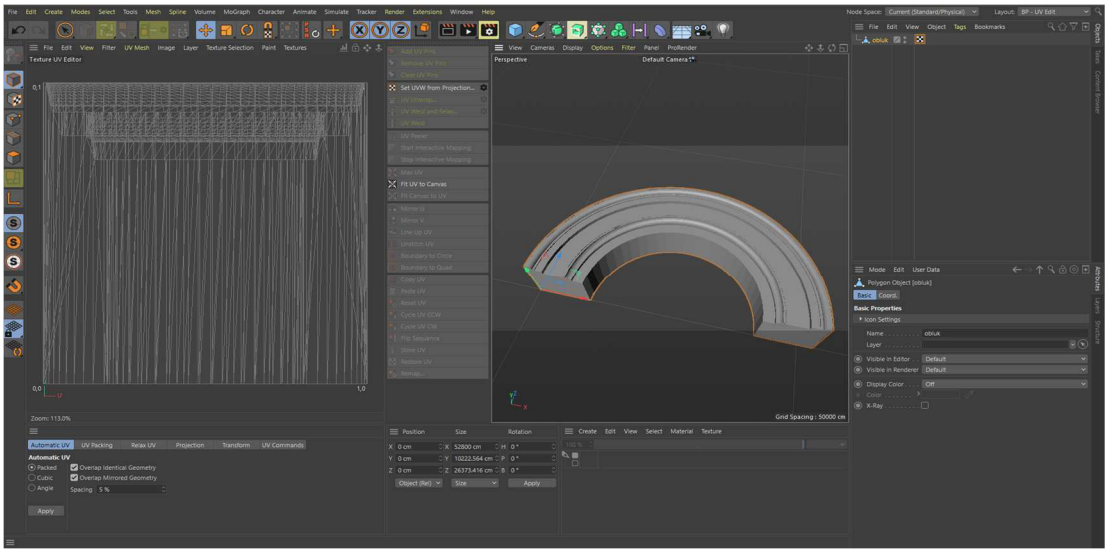

Fig. 5. STL model in Cinema 4D without UV map optimization [6, 8]

As we see, if we import the STL model to the graphics editing software, the model does not have any additional information and the showed UV map on Fig. 5 shows the unoptimized map of the surface, which is created only by interpretation by program of the model. The result is evident on the first sight, if we use the tool for optimizing the UV map.

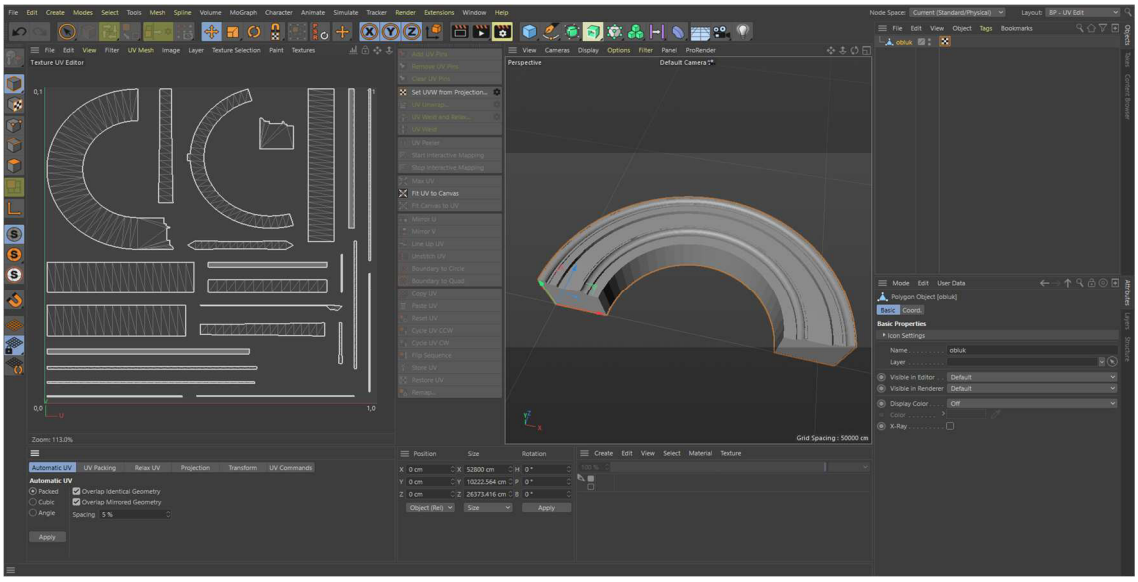

Fig. 6. STL model in Cinema 4D without UV map optimization [6, 9]

We see the end result of the optimized UV map on Fig 6. We used the tool built in the Cinema 4D and exported STL file format as FBX and now we achieved the parity, and our STL model is now FBX and have the information about the surface of the model and how to apply materials to it. On the other hand, if we create the model natively in the editor which is specifically created for graphics design like Blender or MAYA, the UV maps are created or edited in the process of the modeling. But the CAD software is not created for this kind of use case, so that's why we have to find this kind of workaround [10, 11].

The End Result. When we see the end results, it will be pretty similar, which is a good thing. That means that we can use both these workflows, but we need to think about which of these workflows is more efficient for us. The STL convention done manually gives us control on how we want to edit our models, but it requires from us to learn a new software, which is used in game development such as Blender or MAYA or Cinema 4D. On the other hand, the DATASMITH plugin gives us more straightforward conversion and is better for large data imports, such as architectural data.

Fig. 7. STL model and DATASMITH import in Unreal Engine side by side. Right is STL and left is DATASMITH import [6,10]

As we see, they look pretty similar, but we want to point out, that if we choose the STL method we need to take patience with choosing the correct dimensions for our model in CAD software, because if we choose small dimensions and scale it up, we can see aliasing on the edge of right model, Fig. 7. On the other hand, DATASMITH import looks pretty similar but the dimensions also are important, because they can save us time later in the engine, where we don't have to scale them properly later. Thanks to implementation of chord tolerances, the model imported this way is smoothed out in the engine.

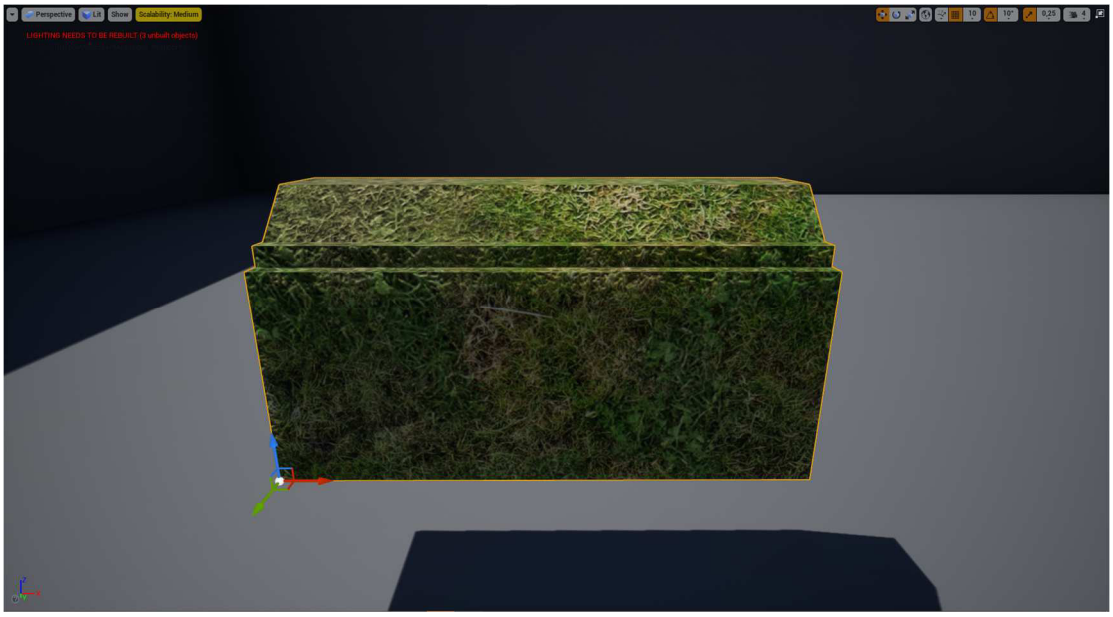

Fig. 8. STL model in Unreal Engine with optimized UV map [6]

If we look at the results of material appliance, the UV map optimization in Cinema 4D or done by DATASMITH is pretty much similar and the editing of the material can be done later, so the end result is very usable and can be used even in industrial applications.

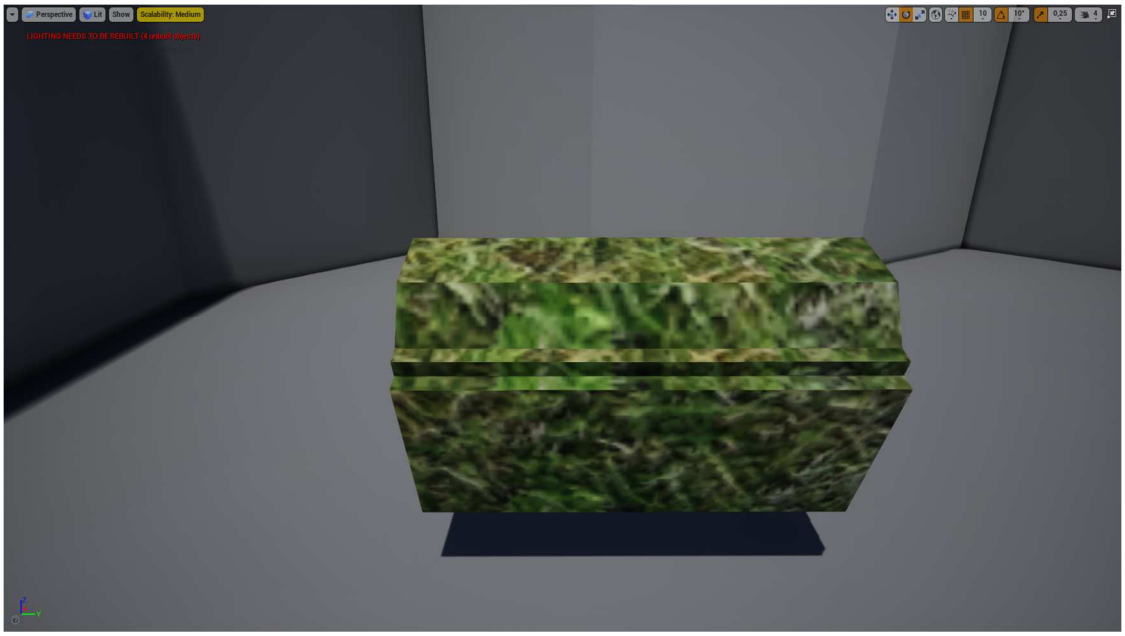

Fig. 9. STL model in Unreal Engine imported by DATASMITH UV map test [7, 9]

Conclusions. As the development of the real time rendering continues and as new technologies such as NVIDIA'S real time raytracing are coming to the mainstream, we are finally hitting that golden standard, which we call photorealism. Which is more impressive, these technologies are accessible through consumer hardware and the performance is doubling in the next generation. As we develop more powerful physics engines and we learn to use raytracing, we can create photorealistic real time simulations in close future, which can give us results in a matter of minutes. Maybe at the start they are not as detailed as traditional CAD software calculations, but with development of these tools we can soon hit that goal too. This opens a lot more possibilities for the future. We can modify the simulation in the middle of calculation or a designer can see in high fidelity detail the final product in real time without the need to wait several hours to render it. Also we can make these tools more accessible for non-professionals and make the learning curve less difficult and save time on rendering. There is also economic value when we can use consumer level hardware to create professional results. These reasons can be the foundation for our prediction that the future of rendering is real time rendering.

Acknowledgement. The authors would like to thank to Slovak Grant Agency – project KEGA 018TUKE-4/2018 "Implementation of new technologies and educational methods in the field of control systems to improve education level and practical skills of graduates of Mechatronics" and project VEGA 1/0389/18, "Research and development of kinematic redundant mechanisms"

References

1. Klein R., Straßer W. (1997) Generation of Multiresolution Models from CAD-Data for Real Time Rendering. In: Strasser W., Klein R., Rau R. (eds) Geometric Modeling: Theory and Practice. Focus on Computer Graphics (Tutorials and Perspectives in Computer Graphics). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-60607-6_21.

2. Y. Tang and H. Gu, "CAD Model's Simplification and Conversion for Virtual Reality," 2010 Third International Conference on Information and Computing, Wuxi, 2010, pp. 265-268, doi: 10.1109/ICIC.2010.338.

3. Ben Abdallah, H.; Jovančević, I.; Orteu, J.-J.; Brèthes, L. Automatic Inspection of Aeronautical Mechanical Assemblies by Matching the 3D CAD Model and Real 2D Images. J. Imaging 2019, 5, 81.

4. H. Li, C. Shen and Q. Shi, "Real-time visual tracking using compressive sensing," CVPR 2011, Providence, RI, 2011, pp. 1305-1312, doi: 10.1109/CVPR.2011.5995483.

5. J. Xue, X. Zhai and H. Qu, "Efficient Rendering of Large-Scale CAD Models on a GPU Virtualization Architecture with Model Geometry Metrics," 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE), San Francisco East Bay, CA, USA, 2019, pp. 251-2515, doi: 10.1109/SOSE.2019.00043.

6. Epic Games, Inc. (2020). Unreal engine learning portal. Retrieved from: https://docs.unrealengine.com/en-US/Engine/Content/Importing/Datasmith/index.html.

7. Epic Games, Inc. (2020). Unreal engine the article about autonomous drives. Retrieved from: https://www.unrealengine.com/en-US/blog/making-autonomous-vehicles-safer-before-they-hit-the-road.

8. Omar Shakshak, Igor Evsikov, Timur Ablyazov, Igor Yamshanov, " Interactive Digital Model of Assessing Energy Efficiency of Buildings," International Conference on Digital Technologies in Logistics and Infrastructure (ICDTLI 2019), Atlantis Press, pp. 2589-4900, doi:10.2991/icdtli-19.2019.59

9. G. Eriksson and A. Engberg, 'Automating the CAD to Virtual Reality Pipeline for Assembly Simulation', Dissertation, 2020.

10. ZHU Hu,YANG Zhongfeng,ZHANG Wei, "Progress in Study of STL File and Its Application," College of Mechanical and Electrical Engineering, Shenyang Institute of Aeronautical Engineering, Shenyang Liaoning 110136,China, http://en.cnki.com.cn/Article_en/CJFDTotal-JCYY200906065.htm.

11. Kenton McHenry, Peter Bajcsy, "An Overview of 3D Data Content, File Formats and Viewers," National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, IL. (2008), Technical Report: isda08-002.

УДК 004.94:004.92

Ерік Прада, Адам Коларж

МОЖЛИВОСТІ КОНВЕРТУВАННЯ CAD-МОДЕЛЕЙ ДЛЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВІЗУАЛІЗАЦІЇ В РЕАЛЬНОМУ ЧАСІ

Актуальність теми дослідження. *Запропоновані методи можуть заощадити дорогоцінний час та в деяких випадках скоротити витрати на обладнання, а також спростити розробку та візуалізацію продуктів для нетехнічних професіоналів.*

Постановка проблеми. *Використання безкоштовних та платних продуктів для застосування в сучасному бізнесі та у школах для оптимізації фази проектування та дослідження, а також створення умов для навчання кваліфікованого персоналу в екстремальних умовах.*

Аналіз останніх досліджень і публікацій. *Ці оптимізації здійснюються безпосередньо розробниками інструментів моделювання в режимі реального часу та деякими великими компаніями, але багато з них розділені на більш дрібні підприємства і потребують одних рекомендацій, які були б основними.*

Виділення недосліджених частин загальної проблеми. *Методологія перетворення CAD моделей для візуалізації в реальному часі.*

Постановка завдання. *Створити єдиний стандарт процедур, що дозволяють використати можливості інструментів візуалізації в режимі реального часу та знайти способи їх використання у виробничій практиці.*

Виклад основного матеріалу. *У статті представлені можливості використання ігрового рушія Unreal Engine та аддону Epic Games DATASMITH для перетворення CAD моделей у програмне забезпечення для візуалізації в реальному часі.*

Висновки відповідно до статті. *Ігрові рушії можна використовувати з такими інструментами, що спрощують художникам та іншим нетехнічним працівникам миттєве бачення результатів роботи. Крім того, ці інструменти можуть бути використані для різних видів моделювання із більш високою ефективністю, але результати можуть бути менш точними. Проте з появою кращих інструментів та кращою інтеграцією програмного забезпечення САПР, ці результати можна покращити. На нашу думку, майбутнім проєктування та тестування продуктів є тестування в реальному часі.*

Ключові слова: *CAD-модель; DATASMITH; Unreal Engine; UV-розгортка; віртуальна реальність.*
Рис.: 9. Бібл.: 11.

Erik Prada – Doctor of Technical Sciences, Proffessor , Technical University of Kosice Department of Mechatronics, Faculty of Mechanical Engineering (Park Komenského 8, 042 00 Košice, Slovak Republic).
E-mail: erik.prada@tuke.sk
ORCID: https://orcid.org/0000-0001-7226-1240
ResearcherID: R-3524-2016
Scopus Author ID: 55873860000

Adam Kolář – Student of Study Program Bachelor of Industrial mechatronic Technical University of Kosice Department of Mechatronics, Faculty of Mechanical Engineering (Park Komenského 8, 042 00 Košice, Slovak Republic).
E-mail: adam.kolar@student.tuke.sk

Prada, E., Kolář, A. (2020). Possibilities of convert cad models for real time rendering software. *Technical sciences and technologies*, *3*(21), pp. 220-228.