UDC 004.946:004.453.4-025.13 **doi:** https://doi.org/10.20998/2522-9052.2025.4.02

Heorhii Kuchuk, Mykyta Matvieiev

National Technical University «Kharkiv Polytechnic Institute», Kharkiv, Ukraine

MODELING THE PROCESS OF LOADING 3D MODELS IN A CLIENT APPLICATION

Annotation. Relevance. The growing use of client-side AR applications increases the demands on the speed of loading and rendering 3D models. Network conditions and the resource constraints of client devices make it necessary to develop simple yet adequate mathematical models of the loading process. Such models enable the identification of bottlenecks and the development of effective optimization methods aimed at reducing waiting time and improving the user experience. **Object of the study:** the process of loading and rendering 3D models in a client application with augmented reality elements. **Purpose of the article:** to develop a simplified mathematical model of the 3D model loading process in the client-side web application implemented with *Angular*, which takes into account key parameters – the size of the model data, network bandwidth, and processing delays – and serves as a foundation for the development of methods to optimize loading time. **Research results.** The study presents a mathematical model that describes the total time T required for loading and rendering 3D models in a client-side AR application. The model formalizes the relationships between the model data size, network bandwidth, processing delays, the overhead constant t_0, and the optimization coefficient δ. The derived analytical dependencies make it possible to evaluate the influence of each parameter on the total time T and to use them as a basis for methods of optimizing the transmission and preprocessing of 3D content. The model has been verified, which confirmed its adequacy within the defined assumptions and parameter ranges. **Conclusions.** A simplified model is proposed that demonstrates both flexibility and ease of application: through the introduction of the overhead constant t_0 and the optimization coefficient δ, it can be adapted to different scenarios of client application operation and varying network conditions. The derived analytical dependencies may be applied in the development of practical optimization methods aimed at reducing the loading time of 3D models and improving the user experience in AR clients. **Scope of application of the obtained results:** client-side AR applications (mobile and web-based), systems for preprocessing and delivery of 3D content, and optimization tools for frontend development projects (including those implemented with *Angular*).

Keywords: 3D model loading; augmented reality; mathematical model; network bandwidth; processing delays; loading time optimization; *Angular*.

Introduction

Problem statement. Interaction with a 3D model, as a new mode of information exchange, provides a more intuitive and efficient representation of data compared to traditional 2D graphics [1]. With the growth of network bandwidth and the increasing computational power of client devices, AR technologies with 3D visualization are being used ever more frequently across various domains – from education to industry [2–5].

Although AR applications offer high performance, their adoption is limited by installation requirements, cross-platform issues, and data compatibility challenges [6, 7]. In contrast, 3D web-applications that run directly in the browser provide maximum portability, rapid access, and versatility, creating a unified foundation for the exchange of 3D content across different devices [8].

However, the implementation of high-quality 3D model rendering in Web AR encounters several challenges. First, the limited resources of client devices do not always permit the processing of large data volumes and the execution of complex computations in real time [9, 10]. Second, 3D model files contain substantially more information – textures, animations, and attribute data, than 2D graphics, which imposes an additional burden on both network and device [11, 12]. The combination of large data volumes and the limited bandwidth of mobile internet leads to loading delays and slower rendering [13]. In response to these issues, the scientific literature widely investigates performance models for web applications, encompassing analytical, empirical, and simulation-based approaches to estimating loading time, the effects of network conditions, and configuration parameters [14–17]. The application of these approaches fosters a deeper understanding of the processes, improves prediction accuracy, and supports optimization of web application performance [8, 18].

Analysis of recent studies and publications. The author of article [19] presents a systematic, step-by-step approach to predicting response time and identifying "bottlenecks" in a distributed web system using modeling. Before building the model, the author clearly defines the key metrics: response time and throughput. Next, a simplified model of the cluster architecture is created and gradually refined. Subsequently, the model is verified by comparing it with measurement results from the real system. The model's average error falls within 10-20%, which is considered acceptable for performance forecasting. Finally, an analysis of various system operation scenarios is carried out to identify potential bottlenecks.

In publication [20] a methodological approach is presented for developing data-intensive applications that run in distributed systems, using modeling and iterative refinement. The proposed method divides the model into functional and operational levels: in the first stage the loading and processing logic is formalized as a functional model, after which network and resource constraints – in particular throughput, service queues, and delays – are taken into account in the operational model. The publication also implements an approach for estimating performance bounds using simulation and approximation, which combines analytical calculation of average latency with experimental modeling to determine the minimum and maximum load times under variable network load.

In [21], the engineers proposed an approach for the step-by-step creation and verification of formal models of large-scale discrete event systems based on Petri nets. First, a simplified conceptual model is constructed and then refined by adding specific components and resource constraints for subsequent automatic transformation of the model into a program for distributed simulation. For each transition, a throughput function is defined that depends on the volume of transmitted data and the number of concurrent requests, in order to reflect the impact of network delays on loading speed.

In publication [22], a real data driven methodology is proposed for constructing and modeling load in a web mapping system. The approach comprises several stages. First, metrics from real user sessions are collected to form an empirical dataset. A session layer is then created: session length is modeled by a log normal distribution, user think times by a Pareto distribution, and pan and zoom operations by a geometric distribution with predetermined probabilities for movement directions. At the operations and data level, each user action corresponds to a specific map tile request that accounts for the volume of transmitted data. The sequence of operations is described using state diagrams based on Markov chains, which specify transition probabilities between different action types. Finally, Monte Carlo simulation is performed with the generated synthetic sessions to validate response time distributions and to compute relevant statistical characteristics.

Although the cited literature is not directly related to modeling the loading time of 3D models, the presented data can be adapted to construct a custom mathematical model of the 3D model loading process in a client application.

The objective of this study is to develop a simplified mathematical model of the 3D model loading process in a client application.

The model will account for the following key parameters: model data size, network throughput, and processing delays. The derived analytical relationships will form the basis for developing optimization methods aimed at reducing loading time and improving the user experience in client AR applications.

Main part

In the context of developing client applications with augmented reality elements, a critical aspect is the optimization of loading and visualization times for 3D models. The performance of such systems directly affects the user experience [23, 24]. This section presents a mathematical model that describes the total time T required to load and display 3D models in a client application with augmented reality elements implemented in Angular.

The model is given by:

$$T = t_0 + \sum_{i=1}^{n} s_i / b \qquad (1)$$

where T is the total loading time in seconds; t_0 is a constant overhead reflecting unaccounted operations and equals 0.5 s; s_i is the size of the i-th model in megabytes; n is the number of models ($1 \le n \le 3$); b is the network throughput in megabytes per second.

The total time to load and display 3D models in the application is denoted as T. According to classical data transfer theory, transmission time depends on data volume and channel throughput [25–27]. The model assumes that loading time is proportional to model size and inversely proportional to network speed. Additional time costs associated with operations not accounted for in the formula, such as environment initialization, parsing and model rendering, are included via the constant t_0.

To validate the model, the dependence of T on Σs_i as computed and the theoretical value T_{theor} obtained from the model was compared with the actual measured time T_{act} in the application. The application throughput was artificially constrained to 1 MB/s using Chrome DevTools. For loading a single 3D model (n = 1) with the following input values:

$$\sum_{i=1}^{n} s_i = 16,4 \text{ MB}; \quad t_0 = 0,5 \text{ s}; \quad b = 1 \text{ MB/s},$$

the theoretical value T_{theor} was calculated using formula (1), then

$$T_{theor} = \frac{16,4}{1} + 0,5 = 16,9(s).$$

Based on the results of three parallel tests, which are presented in Table 1, the average value of T_{act} was calculated.

Table 1 – **Loading time of one model in a web application**

Test	Time, s
1	16,51
2	17
3	16,65
Average value	16,72

According to the results obtained, the difference between T_{theor} = 16,9 s and T_{act} = 16,72 s is 0,18 s which is 1,06% of T_{theor}.

For loading two different 3D models (n = 2) with the following input values:

$$\sum_{i=1}^{n} s_i = 35,1 \text{ MB}; \quad t_0 = 0,5 \text{ s}; \quad b = 1 \text{ MB/s},$$

the theoretical value T_{theor} was calculated using formula (1), then

$$T_{theor} = \frac{35,1}{1} + 0,5 = 35,6(s).$$

Based on the results of three parallel tests, which are presented in Table 2, the average value of T_{act} was calculated.

Table 2 – **Loading time of two models in a web application**

Test	Time, s
1	32,03
2	33,02
3	32,32
Average value	32,45

According to the results obtained, the difference between T_{theor} = 35,6 s and T_{act} = 32,45 s is 3,15 s, which is 8,85% of T_{theor}.

For loading three different 3D models (n = 3) with the following input values:

$$\sum_{i=1}^{n} s_i = 51{,}9 \text{ MB}; \quad t_0 = 0{,}5 \text{ s}; \quad b = 1 \text{ MB/s},$$

the theoretical value T_{theor} was calculated using formula (1) then

$$T_{theor} = \frac{51{,}9}{1} + 0{,}5 = 52{,}4(s).$$

Based on the results of three parallel tests, which are presented in Table 3, the average value of T_{act} was calculated.

Table 3 – **Loading time of three models in a web application**

Test	Time, s
1	44,17
2	43,7
3	43
Average value	43,62

According to the results obtained, the difference between T_{theor} = 52,4 s and T_{act} = 43,62 s is 8,8 s, which is 16,79% of T_{theor}.

The comparison between the theoretical loading time T_{theor} and the actual loading time T_{act} is presented in Fig. 1. The figure shows that as the total size of the elements Σs_i increases, the discrepancy between the theoretical and actual loading times of the 3D model also grows.

The increase in discrepancies between T_{theor} and T_{act} is explained by the fact that the mathematical model does not account for a number of minor optimizations implemented in the web application.

These optimizations include the use of Web Workers, HTTP/2 parallelism, internal browser optimizations, and specific features of the glTF 3D model format [28].

These mechanisms result in the actual loading and rendering time T_{act} being significantly lower than the theoretical estimate T_{theor}. In particular, parallel requests and network-level optimizations typically provide the most substantial acceleration [29].

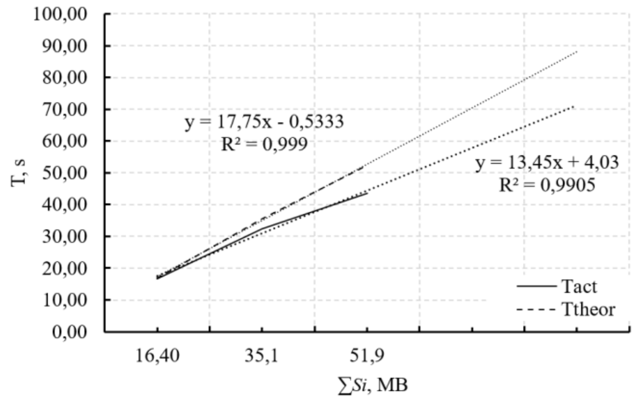

Fig. 1. Dependence of T on Σs_i in the basic model

Based on the above results, it is reasonable to introduce into the model a coefficient $(1 - \delta n)$ that accounts for the effect of optimization. This coefficient reflects the portion of time reduced due to optimizations. It is assumed that as the number of models n increases, the optimization effect also grows, leading to a reduction in the loading time of 3D models.

Taking this coefficient into account, the model takes the following form:

$$T = t_0 + (1 - \delta n) \cdot \sum_{i=1}^{n} s_i / b, \quad (2)$$

where δ is a constant (0.1) that represents an approximate 10% reduction in loading time for each additional model. Since the optimization coefficient depends on the number of models n, the constant t_0, which accounts for unmodeled operations, should be scaled in a similar manner. Moreover, its value should be increased from 0.5 s to 2 s to partially compensate for the overestimated optimization effect at the 10% level.

As a result, the final form of the improved model is expressed as

$$T = n \cdot t_0 + (1 - \delta n) \cdot \sum_{i=1}^{n} s_i / b, \quad (3)$$

Using the new model (3), calculations were performed, the results of which are given in Table 4.

Table 4 – **Dependence of T on $\sum s_i$ in the model (3)**

Number of models, n	T_{theor}, **s**	T_{act}, **s**	Difference**, s**	Difference**, % of** T_{theor}
1	16,76	16,72	0,04	0,24
2	32,08	32,45	-0,37	1,14
3	42,33	43,62	-1,29	3,05

As a result, the new model demonstrated higher calculation accuracy compared to the baseline. The discrepancy did not exceed 5% according to the experimental results, which confirms the high reliability of the model and the slow growth of error with an increase in the total size of the elements (Fig. 2).

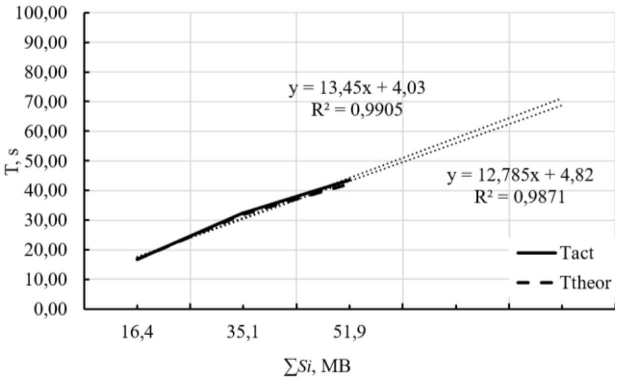

Fig. 2. Dependence of T on Σs_i in the improved model

It is worth noting that in most experiments the improved model shows results where T_{act} slightly exceeds T_{theor}. This provides a margin for further application-level optimization, which would lead to a reduction of T_{act}. The most important advantage of the model lies in its flexibility and simplicity: due to the inclusion of the overhead constant t_0 and the optimization coefficient δ, the model can be adjusted and adapted to potential changes within the client application.

To evaluate the sensitivity of the model, numerical simulations were conducted to analyze the dependence of the total loading time T on variations in the key parameter of network throughput b. Such an analysis makes it possible to determine how strongly the model's results respond to different values of this variable and how this affects the overall performance of the system.

The calculation results are given in Table 5 for the following data:

$$n = 3; \sum_{i=1}^{n} s_i = 51{,}9 \text{ MB}; \quad t_0 = 2 \text{ s}; \quad \delta = 0{,}1.$$

Table 5 – **Dependence of *T* on *b* according to model (3)**

Bandwidth, *b*, MB\s	T_{theor}, **s**	T_{act}, **s**	Difference**, s**	Difference**, % of** T_{theor}
0.5	78,66	78,92	-0,26	0,33
1	42,33	43,62	-1,29	3,05
1,5	30,22	32,38	-2,16	7,15
2	24,17	25,83	-1,66	6,87
2,5	20,53	22,04	-1,51	7,36
3	18,11	19,74	-1,63	9

Based on these data, the dependence of the total time, T, on the bandwidth, b at $\sum s_i$ = 51.9 MB was constructed (Fig. 3).

From Fig. 3 it can be seen that the curves T_{act} and T_{theor} show a rapid decrease in loading time as network throughput increases in the range from 0.5 to 2 MB/s, which is fully expected. After reaching 2 MB/s, with further increases in b, the dependence *T(b)* diminishes – the curves gradually flatten, and the total loading time changes only slightly, mainly due to the influence of the constant overhead t_0 associated with processing each model.

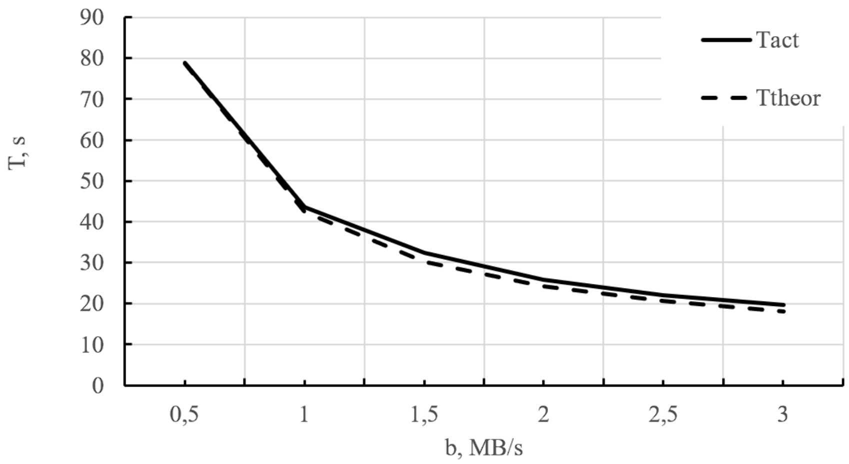

Fig. 3. Dependence of T on b at $\sum s_i$= 51.9MB

This indicates that at high values of network throughput the primary bottleneck is no longer the network itself, but rather client-side data processing, represented by the constant t_0. Therefore, even a twofold increase in throughput from 2 to 4 MB/s yields only a marginal reduction in loading time.

As network throughput increases, T_{theor} decreases significantly, while T_{act} exhibits an increasing deviation from it. This occurs because the model focuses primarily on network delay and accounts only for fixed overhead costs t_0, whereas in the actual application a substantial portion of the time is consumed by client-side operations such as parsing, rendering of 3D models, and initialization of the AR environment – processes that are unaffected by increases in b. As a result, at higher network speeds T_{act} becomes greater than T_{theor}, and the relative error increases. Nevertheless, since the error remains within acceptable limits for applied modeling and is considered reasonable for a simplified model, it was decided not to modify the model further.

Conclusions

A review and analysis of the scientific literature related to the research topic was conducted. A simplified mathematical model of the 3D model loading process in a client application was developed. The model accounts for key parameters: model data size, network throughput, processing delays, and application-level optimizations. The main advantage of the model is its flexibility and simplicity: through the overhead constant t_0 and the optimization coefficient δ, the model can be adjusted and adapted to potential changes. Model verification was carried out.

The obtained analytical relationships provide a foundation for the development of optimization methods aimed at reducing loading time and improving the user experience in client AR applications.

Acknowledgements

The study was funded by the Ministry of Education and Science of Ukraine in the framework of the research project 0125U001544 on the topic "Methodology for ensuring the processes of monitoring and controlling the implementation of project and program portfolios for project offices in the context of Ukraine's reconstruction".

References

1. Sosna, T, Vochozka, V, Šerý, M and Blažek, J (2025), "Developing pupils' creativity through 3D modeling: an experimental study", *Front. Educ*., vol. 10, article number 1583877, doi: https://doi.org/10.3389/feduc.2025.1583877
2. Fernández-Moyano, J. A., Remolar, I. and Gómez-Cambronero, Á. (2025), "Reality's Impact in Industry – A Scoping Review", *Applied Sciences*, vol. 15(5), article number 2415, doi: https://doi.org/10.3390/app15052415
3. Kuchuk, H. and Kuliahin, A. (2024), "Hybrid Recommender For Virtual Art Compositions With Video Sentiments Analysis", *Advanced Information Systems*, vol. 8, no. 1, pp. 70–79, doi: https://doi.org/10.20998/2522-9052.2024.1.09
4. Kuliahin, A., Kuchuk, H. (2023), "Classified emotion as implicit recommendation system feedback", *2023 IEEE 4th KhPI Week on Advanced Technology*, KhPI Week 2023 – Conf. Proc., doi: https://doi.org/10.1109/KhPIWeek61412.2023.10312976
5. Halıcı, S.M. and Gül, L.F. (2025), "AI-based augmented reality for architectural mass study", *Virtual Reality*, vol. 29, article number 146, doi: https://doi.org/10.1007/s10055-025-01142-z
6. Carter, E., Sakr, M. and Sadhu, A. (2024), "Augmented Reality-Based Real-Time Visualization for Structural Modal Identification", *Sensors*, vol, 24(5), article number 1609, doi: https://doi.org/10.3390/s24051609
7. Jeon, J. and Woo, W. (2024), "eCAR: edge-assisted Collaborative Augmented Reality Framework", *arXiv*, arXiv:2405.06872, doi: https://doi.org/10.48550/arXiv.2405.06872
8. Li, L., Qiao, X., Lu, Q., Ren, P. and Lin, R. (2020), "Rendering Optimization for Mobile Web 3D Based on Animation Data Separation and On-Demand Loading", *IEEE Access*, vol. 8, pp. 88474–88486, doi: https://doi.org/10.1109/ACCESS.2020.2993613
9. Kalinin, Y., Kozhushko, A., Rebrov, O., and Zakovorotniy, A. (2022), "Characteristics of Rational Classifications in Game-Theoretic Algorithms of Pattern Recognition for Unmanned Vehicles", *2022 IEEE 3rd Khpi Week on Advanced Technology Khpi Week 2022 Conference Proceedings*, 03-07 October 2022, doi: https://doi.org/10.1109/KhPIWeek57572.2022.9916454
10. Kuchuk, G., Kharchenko, V., Kovalenko, A. and Ruchkov, E. (2016), "Approaches to selection of combinatorial algorithm for optimization in network traffic control of safety-critical systems", *Proceedings of 2016 IEEE East-West Design and Test Symposium*, EWDTS 2016, 7807655, doi: https://doi.org/10.1109/EWDTS.2016.7807655
11. Filatov, V., Filatova, A., Povoroznyuk, A. and Omarov, S. (2024), "Image classifier for fast search in large databases", *Advanced Information Systems*, vol. 8, no. 2, pp. 12–19, doi: https://doi.org/10.20998/2522-9052.2024.2.02
12. Petrovska, I., Kuchuk, H., Kuchuk, N., Mozhaiev, O., Pochebut, M. and Onishchenko, Yu. (2023), "Sequential Series-Based Prediction Model in Adaptive Cloud Resource Allocation for Data Processing and Security", *2023 13th International Conference on Dependable Systems, Services and Technologies*, DESSERT 2023, 13–15 October, Athens, Greece, code 197136, doi: https://doi.org/10.1109/DESSERT61349.2023.10416496
13. Boutsi, A.-M., Ioannidis, C. and Verykokou, S. (2023), "Multi-Resolution 3D Rendering for High-Performance Web AR", *Sensors*, vol. 23(15), article number 6885, doi: https://doi.org/10.3390/s23156885
14. Abdallah, M., Sawalhi, G., Mazhar, A., AlRifaee, M. and Salah, M. (2024), "Factors Influencing the Quality of Augmented Reality Applications", *Procedia Computer Science*, vol. 251, pp. 150–156, doi: https://doi.org/10.1016/j.procs.2024.11.095
15. Ibrahimov, B., Hashimov, E. and Ismayılov, T. (2024), "Research and analysis mathematical model of the demodulator for assessing the indicators noise immunity telecommunication systems", *Advanced Information Systems*, vol. 8, no. 4, pp. 20–25, doi: https://doi.org/10.20998/2522-9052.2024.4.03
16. Zakovorotniy, A., and Kharchenko, A. (2021), "Optimal speed controller design with interval type-2 fuzzy sets", *2021 IEEE 2nd KhpiWeek on Advanced Technology*, pp. 363–366, doi: https://doi.org/10.1109/KhPIWeek53812.2021.9570045
17. Kuchuk, H., Mozhaiev, O., Kuchuk, N., Tiulieniev, S., Mozhaiev, M., Gnusov, Y., Tsuranov, M., Bykova, T., Klivets, S. and Kuleshov, A. (2024), "Devising a method for the virtual clustering of the Internet of Things edge environment", *Eastern-European Journal of Enterprise Technologies*, vol. 1, no. 9 (127), pp. 60–71, doi: https://doi.org/10.15587/1729-4061.2024.298431
18. Pandey, A. and Srivastava T. (2021), "Mathematical Modelling: Growing Role and Applications", *Journal of Applied Science and Education* (JASE), vol. 1, pp. 1–11, doi: https://doi.org/10.54060/JASE/001.01.004

19. Rak, T. (2020), “Modeling Web Client and System Behavior”, *Information*, vol. 11, no. 6, article number 337, doi: https://doi.org/10.3390/info11060337
20. Tolosana-Calasanz, R., Banares, J. A. and Colom, J.-M. (2018), “Model-driven development of data intensive applications over cloud resources”, *Future Generation Computer Systems*, vol. 87. pp. 888–909, doi: https://doi.org/10.1016/j.future.2017.12.046
21. Arronategui, U., Banares, J. A. and Colom, J. M. (2025), “Large scale system design aided by modelling and DES simulation: A Petri net approach”, *Software-Practice & Experience*, vol. 55, no. 2, pp. 243–271, doi: https://doi.org/10.1002/spe.3374
22. Braga, V. G., Correa, S. L., Cardoso, K. V. and Viana, A. C. (2021), “Data-Driven Characterization and Modeling of Web Map System Workload”, IEEE Access, vol. 9, pp. 26983–27002, doi: https://doi.org/10.1109/ACCESS.2021.3058622
23. Kuchuk, N., Kovalenko, A., Kuchuk, H., Levashenko, V. and Zaitseva, E. (2022), “Mathematical Methods of Reliability Analysis of the Network Structures: Securing QoS on Hyperconverged Networks for Traffic Anomalies”, *Lecture Notes in Electrical Engineering*, vol. 831, pp. 223–241, doi: https://doi.org/10.1007/978-3-030-92435-5_13
24. Xie, G., Wang, T., Fu, H, Liu, D., Deng, L., Zheng, X., Li, L. and Liao J. (2025), “The role of three-dimensional printing models in medical education: a systematic review and meta-analysis of randomized controlled trials”, *BMC Med Educ.*, vol. 25, article number 826, doi: https://doi.org/10.1186/s12909-025-07187-7
25. Kuchuk, G.A., Akimova, Yu.A. and Klimenko, L.A. (2000), “Method of optimal allocation of relational tables”, *Engineering Simulation*, vol. 17(5), pp. 681–689, available at: https://www.scopus.com/record/display.uri?eid=2-s2.0-0034512103&origin=resultslist
26. Cardoso, L.F.d.S., Kimura, B.Y.L. and Zorzal, E.R. (2024), “Towards augmented and mixed reality on future mobile networks”, Multimed Tools Appl., vol. 83, pp. 9067–9102, doi: https://doi.org/10.1007/s11042-023-15301-4
27. Kuchuk, H., Husieva, Y., Novoselov, S., Lysytsia, D., Krykhovetskyi, H. (2025), “Load Balancing of the layers Iot Fog-Cloud support network”, *Advanced Information Systems*, vol. 9, no. 1, pp. 91–98, doi: https://doi.org/10.20998/2522-9052.2025.1.11
28. Sahu, D., Nidhi, Prakash, S., Pandey, V. K., Yang. T., Rathore, R. S. and Wang, L. (2025), “Edge assisted energy optimization for mobile AR applications for enhanced battery life and performance”, *Scientific Reports*, vol. 15, article number 10034, doi: https://doi.org/10.1038/s41598-025-93731-w
29. Zhang, H., Li, L., Lu, Q., Yue, Y., Huang, Y. and Dustdar, S. (2024), “Distributed realtime rendering in decentralized network for mobile web augmented reality”, *Future Generation Computer Systems*, vol. 158, pp. 530–544, doi: https://doi.org/10.1016/j.future.2024.04.050

Received (Надійшла) 19.06.2025
Accepted for publication (Прийнята до друку) 10.09.2025

ВІДОМОСТІ ПРО АВТОРІВ / ABOUT THE AUTHORS

Кучук Георгій Анатолійович – доктор технічних наук, професор, професор кафедри комп’ютерної інженерії та програмування, Національний технічний університет “Харківський політехнічний інститут”, Харків, Україна;
Heorhii Kuchuk – Doctor of Technical Sciences, Professor, Professor of Computer Engineering and Programming Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine;
e-mail: kuchuk56@ukr.net; ORCID Author ID: http://orcid.org/0000-0002-2862-438X;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=57057781300.

Матвєєв Микита Іванович – аспірант кафедри комп’ютерної інженерії та програмування, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна;
Mykyta Matvieiev – PhD student, Department of Computer Engineering and Programming, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine;
e-mail: Mykyta.Matvieiev@cit.khpi.edu.ua; ORCID Author ID: https://orcid.org/0009-0000-8773-6640.

Моделювання процесу завантаження 3D моделей у клієнтському застосунку

Г. А. Кучук, М. І. Матвєєв

Анотація. Актуальність. Зростаюче використання клієнтських AR-застосунків підвищує вимоги до швидкості завантаження та відтворення 3D-моделей. Умови мережі та обмеження ресурсів клієнтських пристроїв роблять необхідним створення простих, але адекватних математичних моделей процесу завантаження, які дозволяють аналізувати вузькі місця та розробляти ефективні методи оптимізації для зниження часу очікування й покращення користувацького досвіду. **Об’єкт дослідження:** процес завантаження і відображення 3D-моделей у клієнтському застосунку з елементами доповненої реальності. **Мета статті:** побудова спрощеної математичної моделі процесу завантаження 3D-моделей у клієнтському вебзастосунку, реалізованому на *Angular*, що враховує ключові параметри – обсяг даних моделі, пропускну здатність мережі та затримки при обробці і служитиме підґрунтям для розробки методів оптимізації часу завантаження. **Результати дослідження.** У роботі представлено математичну модель, яка описує загальний час T, необхідний для завантаження та відображення 3D-моделей у клієнтському AR-застосунку. Модель формалізує залежності між обсягом даних моделі, пропускною здатністю мережі, затримками обробки, константою витрат t_0 і коефіцієнтом оптимізації δ. Побудовані аналітичні залежності дозволяють оцінювати вплив кожного параметра на загальний час T і використовувати їх як основу для методів оптимізації передачі та попередньої обробки 3D-контенту. Проведено верифікацію моделі, що підтвердило її адекватність у рамках заданих припущень та діапазонів параметрів. **Висновки.** Запропоновано спрощену модель, яка демонструє гнучкість і простоту застосування: завдяки введенню константи витрат t_0 та коефіцієнта оптимізації δ її можна адаптувати до різних сценаріїв роботи клієнтського застосунку й змін мережевих умов. Отримані аналітичні залежності можуть бути використані для розробки практичних методів оптимізації, які дозволять знизити час завантаження 3D-моделей і покращити користувацький досвід у AR-клієнтах. Сфера використання отриманих результатів: клієнтські AR-застосунки (мобільні та веб-додатки), системи попередньої обробки і доставки 3D-контенту, інструменти оптимізації для розробників фронтенд-проєктів (зокрема на *Angular*).

Ключові слова: завантаження 3D-моделей; доповнена реальність; математична модель; пропускна здатність мережі; затримки обробки; оптимізація часу завантаження; *Angular*.