Romanyuk Oleksandr, Zavalniuk Yevhen, Bobko Oleksii, Stakhov Oleksii, Romanyuk Sergey. Chapter XIII. Artificial intelligence usage for the formation of graphic images. Штучний інтелект у науці : монографія / [авт. колектив]; за ред. Яцишина Андрія та Яцишин Анни. – Київ: ФОП Ямчинський О.В., 2025. – С. 165-181. ISBN 978-617-8830-09-0

CHAPTER XIII. ARTIFICIAL INTELLIGENCE USAGE FOR THE FORMATION OF GRAPHIC IMAGES

DOI: 10.33407/lib.NAES.id/748280

Romanyuk Oleksandr[1[0000-0002-2245-3364]], **Zavalniuk Yevhen**[1[0009-0005-1202-4653]], **Bobko Oleksii**[1[0009-0000-1753-279X]], **Stakhov Oleksii**[1[0000-0002-4901-3211]], **Romanyuk Sergey**[2[0000-0002-6400-0021]]

[1]Vinnytsia National Technical University, Vinnytsia, Ukraine

[2]Odesa National Polytechnic University, Odesa, Ukraine

rom8591@gmail.com

Abstract. In the work, it is analyzed the main aspects of artificial intelligence usage for the formation of graphic images. The main stages of the graphics pipeline for image formation and the prospects of applying artificial intelligence for their optimization are considered. Generative adversarial networks, diffusion models, transformers, ensemble learning methods, which are used for image generation, are characterized. Direct image generation as the alternative to the image formation using graphics editing tools is examined. The ways of optimizing the stages of 3D-model formation, texture generation, texture mapping, 3D-model's surface shading and formed image post-processing are analyzed in detail.

Keywords: rendering, generative adversarial networks, transformers, diffusion model, graphics pipeline.

Introduction. Current state of computer graphics technologies development is characterized by a constant increase of requirements to the realism and productivity of image formation. Formation of highly realistic images involves considering the geometric features of scene's object surfaces, optical properties of materials, key aspects of light interaction with the surface. At the same time, in the systems of virtual reality and interactive computer games it is necessary to provide the formation of images in the real-time mode. The development of artificial intelligence (AI) algorithms has significantly expanded the list of main approaches to image formation and processing. Therefore, AI usage gives the possibility to provide a highly realistic and highly productive image creation at the same time. As a result, the analysis of applying AI for direct image generation and optimization of traditional image formation techniques is actual.

Literature Analysis and Problem Statement

Formation of graphic images [1] is a complex task, which includes a few key stages and uses different technologies for creating, editing, and visualizing the graphic content.

The process begins with the modelling [2], where artists create 3D-models of objects, using special software tools, precisely determining the forms, sizes and proportions of models. The next is texturing [3], where textures are mapped into the surfaces of 3D-models in order to provide them with the realistic appearances. After texturing, the lighting modelling stage begins [4], where the respective shadows and glares on objects are reproduced based on the normalized vectors [5,6] to light source and the viewer. Rendering [7] is the process, where all models, textures and lighting parameters are combined for the creation of final image. As a result, the color intensity is calculated for every point of image. Consequently, rendering can require significant computational resources of computer. There are different techniques of rendering, including rasterization and ray tracing. Rasterization [2] is fast and effective, but cannot provide such level of realism as ray tracing [2], which requires much more computational resources, while providing very realistic images, reproducing complex effects of lighting and shadowing. The process ends with a post-processing [8], which involves color correction, adding visual effects and other changes, which improve the general appearance of the final image.

The listed stages together create the graphics pipeline for image formation (Fig. 1) [9]. The sequence of graphics pipeline stages is general and may vary slightly in special cases.

At the same time, the fast development of AI gives the possibility of optimal object's polygonal model formation, generation of photorealistic textures, determination of appropriate lighting model at the respective stages of graphics pipeline. Additionally, the direct generation of a highly realistic image based on the text description or three-dimensional object's model is possible.

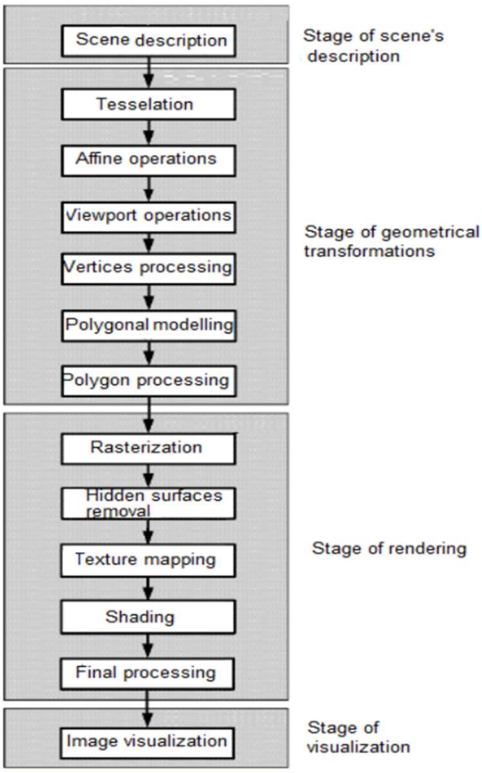

Fig. 1. Main stages of general graphics pipeline [9]

Therefore, AI tools give the possibility to effectively augment the current image formation methods. Hence, conducting the analysis of AI usage for image formation is actual.

Research Results

Main Neural Architectures and Methods for Image Generation

In the process of generating graphic images using AI, various neural network architectures are used, each of which has its own characteristics, advantages, and scope of application.

One of the most common directions is the use of Generative Adversarial Networks (GAN) [10]. This direction involves the interaction of two neural networks - a generator and a discriminator. The generator creates new images based on random input data or feature vectors, and the discriminator tries to distinguish the generated image from the real one. During the training process, both networks are improving, and eventually the generator begins to create images so realistic that the discriminator cannot determine whether they are artificial. Conditional GANs (cGAN) [10] give the possibility to set additional conditions for generation. For example, the user can specify the category of the image. In particular, the Pix2Pix model [11] learns the transformation of an image from one type to another. For example, a satellite photo of a landscape is transformed into a map of the area. At the same time, paired samples of the original and transformed images are necessary during its training. CycleGAN [10], unlike cGAN, does not require using pairs of original and transformed images. The CycleGAN architecture involves the use of two generators and two discriminators. The first generator generates a transformed image based on the original, the second generator reconstructs the original based on the transformed image. As a result, cyclic consistency is ensured. A transformed image is generated in such way that original image can be reconstructed as accurately as possible. Progressive GAN (PGGAN) [10] lies in gradually increasing the sizes of the generator and discriminator during training. This provides a gradual formation of the image from low to high resolution. Wasserstein GAN (WGAN) [10] involves the use of an alternative error function Wasserstein distance, which provides more stable training on the data.

Another type of generative neural network are autoencoders [10], which are used for image compression and restoration. They consist of two parts: an encoder that converts an image into a compressed feature vector, and a decoder that uses this vector to reproduce or generate a new image. Unlike classical autoencoders, Variational Autoencoders (VAEs) work with probability distributions, which allow the model to learn the creation of new images that are variations of those already seen. As a rule, GANs provide clearer images compared to autoencoders.

Another important type of generative neural networks are diffusion models [12], which have become popular thanks to systems such as DALL·E 2, Stable Diffusion, and Midjourney [12]. These models work on the principle of gradual "cleaning" of noise. First, the model takes a random noisy array of pixels, and then, after hundreds or thousands of iterations, using the back diffusion process, transforms it into an ordered image that corresponds to a given text description. This approach ensures high quality and detail of the generated images. Compared to GANs, diffusion models are characterized by more stable training, but less productive image generation.

Another important method is the use of transformers [13], which were originally developed for natural language processing, but later adapted for visual tasks. They provide the possibility to analyze a text query in a complex context, taking into account the semantic connections between words, emotional coloring and style. For example, the CLIP (Contrastive Language–Image Pretraining) model [14], developed by OpenAI, gives the possibility to compare text descriptions with visual images, find correspondences and form semantic connections. Transformers involve the use of attention mechanisms [13], which allow the model to "understand" which elements of the text description are the most important, and accordingly focus attention on generating key details of the image.

The method of image generation through combining several layers of different levels of abstraction — the so-called Hierarchical Generative Models — is gaining popularity. Image creation occurs in stages: first, the general structure of the scene is formed, then the main color arrays are filled, and only after that, small details, shadows, textures, and realistic effects are added.

In addition, there are ensemble methods that combine several models simultaneously [15]. In such systems, the result is generated by several different models, after which the best option is selected or several partial results are combined into a single composition. In particular, Y. Wang et al. [15] distinguish three directions for training GAN ensembles. Standard ensemble (eGANs) [15] lie in that the several GANs are trained on a dataset with random initial parameters. When generating an image, one of the models is randomly selected. Self-ensemble GANs (seGANs) [15] lie in that the each GAN in the ensemble is characterized by the same set of initial parameters and differs in the number of training iterations. Cascade GANs (cGANs) [15] are characterized by the use of a sequence of GANs. Each GAN improves the visualization results of the previous model, and the transition to the next GAN is carried out if the discriminator accuracy exceeds a certain threshold value.

One of the most promising and relatively new directions is the use of hybrid models that combine the capabilities of language and image processing in a single architectural space. They are able to analyze text queries more deeply, taking into account not only keywords, but also context, logic, stylistic features and even the hidden mood of the description. An example of a hybrid model is the combination of the CLIP model, which provides the analysis of correspondence between text and image, and VQGAN for image generation [16]. VQGAN (Vector Quantised GAN) is a modification of GAN, which lies in using a special codebook to improve the quality of the generated image.

Another promising direction is the use of reinforcement learning, when the model learns to improve results not only based on statistical correspondences, but also through "rewards" for high-quality images.

Key Aspects of Training Model for Image Generation

An important aspect of the process of generating graphical images using AI is training the model. To achieve high-quality generation, neural networks are trained on huge arrays of images accompanied by descriptions, tags or metadata. The model gradually learns to recognize patterns, structures, color schemes, compositional techniques and contextual connections between words and visual elements. For example, it remembers that the phrase “blue sky” is most often associated with a background, and “person in a coat” with a certain type of clothing. The more data the model receives, the more accurately it can generate relevant images. However, training on such arrays also creates the risk of accidentally reproducing fragments of existing images.

In addition, modern systems often use the Transfer Learning method - retraining already trained models on new, more specific data sets. This gives the possibility to adapt models to specific styles and genres. As a rule, transfer learning is used in cases of limited data, high complexity of training a model from scratch. An important innovation is LoRA (Low-Rank Adaptation) - a method of retraining models with minimal computing resources. The method lies in "freezing" the training parameters of a large model and training the parameters of additional small-sized matrices.

Stages of the Direct Image Generation Process

The process of generating graphic images using AI consists of several sequential stages that depend on the type of model used, but the general logic of such systems remains the same.

It all starts with entering a text description or other type of query — it can be a key phrase, a set of tags, an image, or a combination of them. Such a text query is called a “prompt” and is the basis for further work. The user formulates his vision of the desired result: for example, “a futuristic city at night, with neon lighting, in a cyberpunk style.”

The AI-system then interprets this query using a pre-trained transformer model that converts the text into a numerical representation—a feature space vector. This vector contains semantic information about objects, their relationships, styles, colors, spatial location, and more.

The next stage is the image generation itself. In the case of GAN [10], a special generator creates an initial image based on feature vectors, after which the discriminator evaluates its realism. Both networks work together until the generator learns to create the most plausible images. In diffusion models [12], the process begins with random noise — a matrix of random pixels, which is cleaned up in several hundred or thousand iterations and takes the form of an image according to the content of the query. The model constantly compares the intermediate results with the semantics of the entered description, adjusting the shape, color, shadows, proportions of objects and other visual characteristics. Fig. 2 shows an example of an image generated using the DALLE 3 model.

Once the image is generated, the user has the opportunity to modify it. In particular, the Neural Style Transfer method can be used to change the style. The neural network extracts contour or structural information from one image and style information from another, and then combines them into a new visual object. For example, images are transformed into "paintings" in the style of Van Gogh, Picasso, or modern artists.

Fig. 2. Example of generated image using DALLE 3 model

From a technical perspective, the entire image generation process is implemented on powerful graphics processing units (GPUs), which process large amounts of data and perform parallel calculations necessary to quickly generate high-quality images. The result is output as a finished digital image in JPEG, PNG or even PSD format.

Applying Artificial Intelligence at Different Stages of Graphics Pipeline

Let's analyze the application of AI at the following stages of the graphics pipeline [9]: polygonal model formation, texturing, shading, and post-processing.

Using AI to generate polygonal models is one of the promising technologies in 3D-graphics. A polygonal model is a 3D-model built from a large number of polygons that describe the surface of an object. Typically, creating such models requires complex manual labor and a large amount of time, but applying AI significantly optimizes this process.

One of the main directions of appying AI in creation of polygonal models is generation of three-dimensional objects on the basis of two-dimensional images. For example, AI models built on the basis of convolutional neural networks (CNN) can restore depth of a scene using so-called depth maps, which are then used to build a polygonal mesh. Another promising method is to use generative models to form 3D models based on text description. For example, DreamFusion [17] from Google allows to create detailed 3D model on the basis of given text. In Fig. 3 examples of DreamFusion generated 3D models based on text description are given.

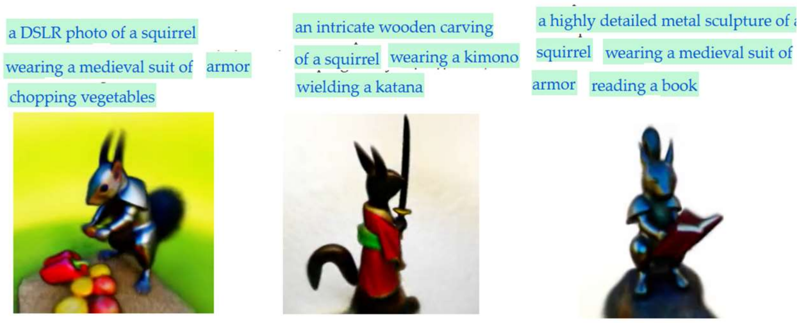

Fig. 3. Examples of generated 3D-models using DreamFusion [17]

Another direction is automated retopology [18]. This is the process of transforming an irregular polygonal model into an ordered, optimized model with fewer polygons [19] using AI or specialized algorithms, while preserving the object's shape and visual quality. Retopology is an important step in 3D-modeling, especially when the original model was created manually or generated by scanning or other methods, which often produce a large number of triangles without a logical structure. Such "dirty" geometry is not suitable for animation, real-time rendering, or integration into game engines. Modern automated retopology tools, such as ZRemesher in ZBrush, Quad Remesher, Instant Meshes, TopoGun, as well as individual AI solutions built into Blender, Houdini, and Maya, use a variety of algorithms: from heuristic and analytical to neural networks. The user can set parameters such as density limits and level of detail. More advanced algorithms are able to recognize anatomical features or logical parts of an object and independently determine where a higher mesh density is needed (for example, in places of animation), and where it can be reduced without loss of quality. In particular, scientists from the University of Berkeley (USA) are developing the Retopokill software tool [18] for optimizing polygonal models in Blender based on AI.

Let’s consider a conceptual model of automated AI retopology of object models. First, the AI analyzes the surface of the object, identifying the main features of the shape: bends, protrusions, depressions. The system builds a surface curvature map, which determines where more polygons need to be placed to maintain accuracy, and where the mesh density can be reduced. The second stage is to determine the edge flow and guide curves of the model. The neural network analyzes the logic of building the topology: for example, how to place polygons around the eyes, mouth, joints, folds of the object. The third stage is building the primary mesh. At this stage, the system forms a base mesh of regular quads or triangles. The algorithm automatically adjusts the sizes and locations of polygons to preserve the general contour of the surface, while minimizing the number of polygons and preventing the appearance of excess geometric elements. The next stage is to optimize the newly created topology. The algorithm corrects the placement of vertices, eliminates topological artifacts (for example, excess vertices). The mesh density can be corrected - for example, higher density in areas of the face where there will be deformation during animation, and lower in flat areas, such as the back. After that, attributes from the initial model are transferred - such as normal maps, textures. The final stage is the output of the final retopologized model, ready for export. The user is given the opportunity to compare the initial and optimized models, as well as manually make adjustments if necessary. In general, the direction of automated retopology is characterized by a significant amount of research, but the number of ready-made solutions in the form of plugins is insignificant. This is explained by the instability of the results of AI optimization of the polygonal model, imperfect training data sets.

AI can automate the texturing process by using large datasets to generate textures from existing images, which significantly speeds up and optimizes the workflow. The Texture Synthesis method [20] allows using the VGG-19 (CNN) model to generate a texture that matches a real photo. To determine the accuracy of texture generation, the Gram matrices for the reference and generated images are compared. Technologies such as GAN are used to increase the resolution of textures. In addition, AI can detect and automatically correct visual defects in textures.

The shading stage involves determining the color intensity of each pixel in the image. In computer graphics, there are a large number of shading methods [1], such as Gouraud shading, Phong shading, Flat shading, PBR shading, and others, which differ in the level of realism, performance, and stylization. The choice of a shading method usually depends on the context of the task: the scene, the material, the target graphics style (realistic or stylized), technical constraints, and the target device (game console, mobile device, render farm). AI can be effectively used to automatically select the shading method, which is especially useful in interactive rendering systems.

There are four directions of using AI to select a shading method for scene. The first method is to classify the scene by type. The neural network analyzes the input parameters (number of objects in the frame, presence of dynamic lighting, type of materials, type of game or application) and classifies it as corresponding to a certain style: realistic, stylized, technical, artistic, etc. Based on the analysis, the AI recommends the optimal shading method: for realistic scenes - PBR or Phong shading, for stylized - Toon shading, for prototype scenes or scenes of technical renderers - Flat shading. The second direction is the analysis of materials and lighting. The AI system can study the materials of the scene, their characteristics (surface shininess, roughness), the number of light and shadow sources, and on this basis predict which shading method will best convey the visual properties of the material. For example, if many shiny surfaces are detected, AI can recommend using Phong shading, if matte materials prevail, Gouraud shading, and in the case of complex materials (hair, fabric) - PBR shading. The third direction is performance optimization. It is possible to build a system that analyzes device resources, the number of objects in the scene, the frame rate and dynamically changes the shading method in real time. For example, on weak devices, it automatically turns on Flat shading or Gouraud, and on powerful devices, PBR with HDR lighting. The fourth option is to use generative models that, based on text queries, automatically "guess" which shading method best suits a given artistic style. For example, if the user enters the prompt "in the style of Japanese animation", AI applies Toon shading, and if the style is "post-apocalyptic photorealism", it selects PBR shading.

In addition to the choice of the surface shading method, the choice of the bidirectional reflectance distribution function (BRDF) [21-24] is important, which is used, in particular, to reproduce glare on object surfaces. BRDF is a key element in rendering, as it describes how light is reflected from a surface depending on the angle of incidence and observation. Different materials have their own unique reflection properties - for example, metal has a strong specular reflection, plastic - combined, and fabric - mainly diffuse. Traditionally, the selection of an appropriate BRDF model (e.g., Phong, Cook-Torrance) and adjustment of its parameters is performed manually, which requires a lot of experience. AI gives the possibility to automate this process. The process of selecting a BRDF using the AI method begins with the analysis of input data - this can be photographs or scanned data of the object, a 3D model, or even a text description of the material. A neural network trained on a large database of examples automatically classifies the type of material, detects its properties (roughness, surface shininess) and matches them with BRDF models that best reflect the physical behavior of this material. For example, the system can automatically recognize that a shiny black surface is a lacquered plastic and select the appropriate BRDF with a microfacet structure.

Another direction is the use of neural networks to model unique BRDFs that cannot be described by classical analytical methods. Typically, such functions provide a compact approximation of measured data sets of the reflectance of materials. For example, the neural network proposed by Sztrajman et al. [25] takes as input the halfway and difference vectors, models the bidirectional reflectance distribution function and provides prediction of the RGB components of the color intensity. The calculation time of the neural network BRDF is comparable to the calculation time of the analytical BRDFs. Only 675 parameters are stored. In the standard case, this would require storing a large table of measured data and applying decomposition methods to it.

In addition, modern AI systems can not only select a BRDF model, but also optimize its parameters, such as roughness, shininess, albedo. This is achieved through inverse rendering methods. A common direction of inverse rendering is the use of neural radiance fields (NeRF) [26-28], which lie in predicting for selected points radiance values and volumetric densities based on input photos. Based on the predicted data, volumetric rendering methods are applied and the final image is formed. Similarly, specialized neural rendering methods have been proposed for the reconstruction of BRDF values. The NeRO method [29] provides the reconstruction of surface geometry and BRDF values by directly using the rendering equation, as well as separate multilayer perceptrons for modeling direct and indirect illumination.

When rendering 3D scenes, it is useful to select the optimal colors of materials. For example, cold colors of materials can convey the atmosphere of the evening, and warm ones - a cozy environment. AI can select the color for objects depending on their role, the general atmosphere of the scene and lighting. If the object should attract attention, AI will choose more saturated, contrasting colors; if the object is secondary – muted, background colors. When selecting colors, AI systems can use the rules of color science (color circles, contrast and complement theories), as well as databases of trend combinations - for example, from Pinterest, Behance or Adobe Color.

The post-processing stage of the generated image is also characterized by the wide possibilities of using AI. In particular, AI is used for anti-aliasing, improving texture detail level, correcting lighting and shadows, and changing the image style. Anti-aliasing is the process of smoothing out the jagged edges that appear when generating images with limited resolution of the display device. Traditional anti-aliasing methods (SSAA, MSAA, FXAA, TAA) cope with this task quite well, but have their limitations: high load on the GPU, image blurring, problems with details or motion effects. The use of AI gives the possibility to achieve better anti-aliasing quality with lower resource consumption and with a smarter approach to preserving details.

Let's consider AI-based anti-aliasing methods: DLSS, DLAA, FSR, XeSS, as well as the new concept of Smart Anti-Aliasing.

A well-known example of combining SSAA (Super Sampling Anti-Aliasing) and AI is Nvidia's Deep Learning Super Sampling (DLSS) technology. DLSS [30] lies in that the deep neural network is trained on the pairs "low resolution + artifacts" - "high resolution without aliasing". During rendering, the image is formed in low resolution and intelligently scaled.

Another technology Nvidia DLAA (Deep Learning Anti-Aliasing) [30], unlike DLSS, involves preserving the original resolution. A special intelligent technology is used to smooth the contours. In general, the image processing process is of higher quality, but less productive compared to DLSS.

Intel XeSS technology [31] uses machine learning for smoothing and scaling, working even on third-party video cards.

AMD FidelityFX Super Resolution (FSR) [32] 4 also provides increased image resolution and is evolving towards integration with AI.

Traditional Temporal Anti-Aliasing (TAA) uses information from previous frames to smooth the image, but has drawbacks such as motion blur. AI can analyze frames more deeply and adapt anti-aliasing much more accurately. For example, K. Herveau et al. [33] proposed using CNNs to predict the values of two filters used to combine the rendered frame and information from previous frames.

Let's analyze the concept of Smart Anti-Aliasing. In future implementations, AI will be able to understand what is depicted in the scene. For example, if an edge belongs to a character's face, one type of anti-aliasing is applied, and if it is part of the distant background, another, less expensive type.

This is a step towards fully integrating AI into graphics pipelines.

Let's compare classical and AI anti-aliasing methods (Table 1) according to the criteria of "productivity", "smoothing quality", "detail preservation", "dependency on resolution", "scene adaptation", "training on data", "GPU support", "application".

Table 1

Comparison of classical and AI-based anti-aliasing methods

Characteristics	Classical methods (SSAA, FXAA, MSAA, TAA)	AI-based methods (DLSS, DLAA, XeSS, FSR)
Productivity	Average or low	High
Smoothing quality	Good, but often with artifacts (blur, ghosting)	High accuracy, minimum artifacts
Detail preservation	Fine textures are often blurred	The sharpness of edges and details is preserved
Dependency on resolution	Work better when the screen resolution is high	Efficient even when screen resolution is relativelty low
Scene adaptation	Fixed approach	Dynamic adaptation to the motion and context of scene
Training on data	Absent	Big datasets are used
GPU support	High compability	Require special hardware (like tensor cores) and API
Application	Universal (games, UI, video)	Mainly in games with AI support

The advantages of AI anti-aliasing are higher quality of smoothing, which is performed only where needed, fewer artifacts, and adaptation to the scene context.

Let's consider the main modern AI anti-aliasing methods in more detail (Table 2).

Table 2

Examples of modern AI-based methods of anti-aliasing

Title	Developer	Technology	Characteristics	Accessibility
DLSS 4	NVIDIA	Super Sampling+AI	Frame Generation, high-quality smoothing	RTX 50, partially RTX 20+
DLAA	NVIDIA	AI Anti-Aliasing	Focused on quality without scaling	RTX 20+
XeSS	Intel	AI Upscaling	Performs best with Intel Arc, supports other GPUs	Intel Arc, Xe-LP, AMD, NVIDIA
FSR 4	AMD	AI Upscaling	Introducing AI-upscaling for a high quality smoothing	AMD RX 9070/ 9070 XT

Therefore, AI upscaling is broadly used to increase the resolution and quality of images. Rendering at low resolution before upscaling provides efficient generation of 3D-frames.

Conclusions. In the work, it is analyzed the usage of AI for generating graphic images. Diffusion models, GANs, and transformers are most often used to generate images. Transformers effectively analyze the semantics of a text query. GANs generate realistic images, but their training is unstable. Diffusion models generate high-quality images and are stably trained, but are less productive. Direct image generation significantly saves image creation time compared to using traditional graphic tools. The use of AI at different stages of the graphics pipeline gives a possibility to generate a 3D-model and texture from a photo, optimize a 3D-model, select a shading method, encode tabular values of the light reflection model, select material colors, and improve the quality of image post-processing.

References

1. Романюк, О. Н., Романюк, О. В., & Чехместрук, Р. Ю. (2023). *Комп'ютерна графіка*. ВНТУ.

2. Akenine-Möller, T., Haines, E., & Hoffman, N. (2018). *Real-Time Rendering, Fourth Edition*. CRC Press LLC.

3. Ebert, D. S., Musgrave, F. K., Peachey, D., Perlin, K., & Worley, S. (2002). *Texturing & Modeling: A Procedural Approach, Third Edition*. Morgan Kaufmann.

4. Birn, J. (2006). *Digital Lighting and Rendering*. Pearson Education, Limited.

5. Romanyuk, O., Zavalniuk, Y., Bobko, O., Arsenyuk, I., Romanyuk, O., Sachaniuk-Kavets'ka, N., Nykyforova, L., Kotyra, A., Kozbakova, A., & Kalizhanova, A. (2024). Method of combined vector normalization of 3-D objects. In *Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2024* (Art. no. 134000O). SPIE.

6. Romanyuk, O., Zavalniuk, Y., Mykhaylov, P., Chekhmestruk, R., Titova, N., & Romanyuk, S. (2024). Accelerated Vector Normalization for Rendering Tasks. In *2024 14th International Conference on Advanced Computer Information Technologies (ACIT)* (pp. 640–644). IEEE. https://doi.org/10.1109/acit62333.2024.10712602.

7. Pharr, M., Humphreys, G., & Jakob, W. (2016). *Physically Based Rendering: From Theory to Implementation*. Elsevier Science & Technology Books.

8. Blinn, J. (1996). *Jim Blinn's corner: A trip down the graphics pipeline.* Morgan Kaufmann Publishers.

9. Romanyuk, O. N., Bobko, O. L., Zavalniuk, Y. K., Titova, N. V., Romanyuk, S. O., & Stakhov, O. Y. (2024). Analysis of graphics pipelines stages. In *Innovation in modern science ‘ 2024* (pp. 71–80). ScientificWorld-NetAkhatAV.

10. Langr, J., & Bok, V. (2019). *GANs in action*. Manning.

11. Завальнюк, Є. К., Романюк, О. Н., & Коробейнікова, Т. І. (2023). Високопродуктивне нейромережеве зафарбовування тривимірних фігур на основі моделі Pix2Pix. *Прикладні питання математичного моделювання*, 6(1), 54–65.

12. Das, D. (2023, October 21). *How does MidJourney create images in real time? Medium*. https://medium.com/hackrlife/how-does-midjourney-create-images-in-real-time-a07fad2df3da

13. Vaswani, A., Shazeer, N., Parmar, N., Uzkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. In *NIPS 2017* (pp. 6000–6010).

14. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning* (pp. 8748–8763).

15. Wang, Y., Zhang, L., & van de Weijer, J. (2016). Ensembles of Generative Adversarial Networks. In *Workshop on Adversarial Training, NIPS 2016*.

16. Crowson, K., Biderman, S., Kornis, D., Stander, D., Hallahan, E., Castricato, L., & Raff, E. (2022). VQGAN-CLIP: Open Domain Image Generation and Editing with Natural Language Guidance. In *Computer Vision – ECCV 2022* (pp. 88–105).

17. Poole, B., Jain, A., Barron, J. T., & Mildenhall, B. (2022). *DreamFusion: Text-to-3D using 2D Diffusion*. DreamFusion3D. https://dreamfusion3d.github.io/

18. AI Retopology for 3D Modeling. (2025). *Alpha3D*. https://www.alpha3d.io/kb/3d-modelling/ai-retopology/

19. Romanyuk, O., & Zavalniuk, Y. (2024). Deep learning-based determination of optimal triangles number of graphic object’s polygonal model. In *IntelITSIS’2024* (pp. 39–51). CEUR.

20. Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). Texture synthesis using convolutional neural networks. In *NIPS'15* (pp. 262–270). MIT Press.

21. Romanyuk, O., Zavalniuk, Y., Pavlov, S., Chekhmestruk, R., Bondarenko, B., Koval, T., Kalizhanova, A., & Iskakova, A. (2023). New surface reflectance model with the combination of two cubic functions usage. *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*, 13(3), 101–106.

22. Romanyuk, O., Zavalniuk, Y., Romanyuk, O., Snigur, A., Titova, N., & Maidaniuk, V. (2023). The development of physically correct reflectance model based on

logarithm function. In *2023 13th international conference on advanced computer information technologies (ACIT)* (pp. 483–487). IEEE.

23. Romanyuk, O., Zavalniuk, Y., Titova, N. V., Kaduk, O., Wójcik, W., Kalimoldayev, M., & Shermantayeva, Z. (2023). Optical system visualization of combined reflectance model based on cubic and quadratic functions. In *Optical fibers and their applications 2023*. SPIE.

24. Romanyuk, O., Lytvynenko, V., & Zavalniuk, Y. (2025). Modified cosine-quadratic reflectance model. *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*, 15(1), 45–48.

25. Sztrajman, A., Rainer, G., Ritschel, T., & Weyrich, T. (2021). Neural BRDF Representation and Importance Sampling. *Computer Graphics Forum*, 40(6), 332–346.

26. Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2022). NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Communications of the ACM*, 85(1), 99–106.

27. Romanyuk, O., Zavalniuk, E., Korobeinikova, T., Titova, N., & Romanyuk, S. (2021). The overview of neural rendering. *Modern engineering and innovative technologies*, Issue № 27, Part 1, 129–134.

28. Завальнюк, Є. К., & Романюк, О. Н. (2024). Використання штучного інтелекту в тривимірній графіці. У *Штучний інтелект у науці та освіті (AISE 2024)* (с. 364–366). УкрІНТЕІ.

29. Liu, Y., Wang, P., Lin, C., Long, X., Wang, J., Liu, L., Komura, T., & Wang, W. (2023). NeRO: Neural Geometry and BRDF Reconstruction of Reflective Objects from Multiview Images. *ACM Transactions on Graphics*, 42(4), 1–22.

30. *NVIDIA DLSS 4*. (2025). NVIDIA. https://www.nvidia.com/en-us/geforce/technologies/dlss/

31. *What is Intel XeSS ?* (2025). Corsair. https://www.corsair.com/us/en/explorer/gamer/gaming-pcs/what-is-intel-xess/

32. warren_eng. (2025). *Game-Changing Updates: FSR 4, AFMF 2.1, AI-Powered Features & More!* Community.AMD.https://community.amd.com/t5/gaming/game-changing-updates-fsr-4-afmf-2-1-ai-powered-features-amp/ba-p/748504

33. Herveau, K., Piochowiak, M., & Dachsbacher, C. (2023). Minimal Convolutional Neural Networks for Temporal Anti Aliasing. In *High-Performance Graphics* (p. 33–41).

* * *