ЕКСПЕРТНА ДУМКА

наукове видання

EXPERT OPINION

scientific publication

№2

isst970@gmail.com
expert-opinion.pp.ua

SCIENTIFIC PUBLICATION

SCIENTIFIC PUBLICATION

«EXPERT OPINION»

ISST

SCIENTIFIC PUBLICATION «EXPERT OPINION»

The scientific publication is containing scientific articles by Doctors and Candidates of Sciences, Doctors of Philosophy and Arts, postgraduate students, applicants for higher (professional pre-higher) education, student researchers and practical worker from European and other countries. The articles contain research that reflects current processes and trends in the development of world science.

Recommended for printing and publication on the Internet
(Decision № 15-2025/1)
www.expert-opinion.pp.ua

Published by Staten House LLC

©Text Copyright 2025 by the International Center for Science and Social Transformation (www.expert-opinion.pp.ua) and authors.
©Cover art: International Center for Science and Social Transformation (www.expert-opinion.pp.ua).
©All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, distributed, or transmitted, in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher. When using and borrowing materials, reference to the publication is required.

Reviewers: Olena ARSHA, Maksym KAISTRO

The texts of scientific materials are published in the author's version with minor corrections (without changing the meaning). The authors and their supervisors are responsible for the accuracy of the information provided.

UDC 001.1

ISBN – 979-8-90046-510-4

The materials are approved by the scientific and metrological base:









TABLE OF CONTENT

ВЗАЄМОЗВ'ЯЗОК ЖИТТЄСТІЙКОСТІ ТА АДАПТИВНОСТІ УЧАСНИКІВ БОЙОВИХ ДІЙ В УКРАЇНІ 7

СТАТЕВА ДИФЕРЕНЦІАЦІЯ ВІДЧУТТЯ ОСОБИСТОЇ БЕЗПЕКИ В УМОВАХ ВІЙНИ 16

СОЦІАЛЬНО-ПСИХОЛОГІЧНІ ПЕРЕДУМОВИ РЕАБІЛІТАЦІЇ ВІЙСЬКОВОСЛУЖБОВЦІВ 25

СПЕЦИФІКА ФУНКЦІОНУВАННЯ МЕХАНІЗМІВ ПСИХОЛОГІЧНОГО ЗАХИСТУ У ФАХІВЦІВ СОЦІОНОМІЧНИХ ПРОФЕСІЙ 36

СОЦІАЛЬНО-ПСИХОЛОГІЧНИЙ ТРЕНІНГ РОЗВИТКУ ПОЗИТИВНОЇ АДАПТАЦІЇ ВЕТЕРАНІВ 46

ПСИХОЛОГІЧНА АДАПТАЦІЯ ОСОБИСТОСТІ В УМОВАХ КРИЗИ: ВПЛИВ ДОСВІДУ БОЙОВИХ ДІЙ ТА ОКУПАЦІЇ 55

ТЕОРЕТИКО-ПРАКТИЧНІ ЗАСАДИ ФОРМУВАННЯ ЧИСЕЛЬНОСТІ ПРАЦІВНИКІВ АГРОПРОМИСЛОВОГО КОМПЛЕКСУ УКРАЇНИ 64

РАЦІОНАЛЬНЕ ВИКОРИСТАННЯ ЗЕМЕЛЬНИХ РЕСУРСІВ: БАЛАНС ЕКОНОМІЧНОЇ ВИГОДИ ТА СОЦІАЛЬНО-ЕКОЛОГІЧНОЇ ВІДПОВІДАЛЬНОСТІ 75

МЕТОДИ ПОКРИТТЯ ФІНАНСОВИХ РИЗИКІВ ПІДПРИЄМСТВА З МЕТОЮ ЗАПОБІГАННЯ ДЕФОЛТУ 84

ПОЗАУРОЧНА ДІЯЛЬНІСТЬ З МАТЕМАТИКИ ЯК ЗАСІБ РОЗВИТКУ ПІЗНАВАЛЬНОГО ІНТЕРЕСУ В УЧНІВ ПОЧАТКОВИХ КЛАСІВ 94

ТЕОРЕТИКО-МЕТОДИЧНІ ОСНОВИ ФОРМУВАННЯ МОВНОЇ КОМПЕТЕНТНОСТІ ФАХІВЦІВ СФЕРИ ДЕРЖАВНОГО УПРАВЛІННЯ 102

МОДЕЛЮВАННЯ ЯК ОДИН З МЕТОДІВ РОЗВИТКУ ПРОСТОРОВОГО МИСЛЕННЯ НА УРОКАХ ІНФОРМАТИКИ В ПОЧАТКОВИХ КЛАСАХ 106

ПРО ДЕЯКІ ПЕДАГОГІЧНІ ЗАСОБИ СТИМУЛЯЦІЇ МОТИВАЦІЇ ДО НАВЧАННЯ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ 112

ФОРМУВАННЯ ДОСЛІДНИЦЬКИХ УМІНЬ УЧНІВ ПОЧАТКОВИХ КЛАСІВ В УМОВАХ STEM-ОСВІТИ 117

МЕТОДИКА І СПЕЦИФІКА РОБОТИ ЕКСПЕРИМЕНТАЛЬНОЇ МАЙСТЕРНІ ТЕАТРАЛЬНИХ ДОСЛІДЖЕНЬ (ЕКМАТЕДОС) 125

ВІДШКОДУВАННЯ МОРАЛЬНОЇ ШКОДИ, ЗАВДАНОЇ КРИМІНАЛЬНИМ ПРАВОПОРУШЕННЯМ, А ТАКОЖ СПРИЧИНЕНОЇ

НЕЗАКОННИМИ ДІЯМИ ОРГАНІВ, ЩО ЗДІЙСНЮЮТЬ ОПЕРАТИВНО-РОЗШУКОВУ ДІЯЛЬНІСТЬ, ОРГАНІВ ДОСУДОВОГО РОЗСЛІДУВАННЯ, ПРОКУРАТУРИ І СУДУ 132

ПРАВО НА ЖИТЛО ЯК ЕЛЕМЕНТ СОЦІАЛЬНОЇ ІНТЕГРАЦІЇ ВНУТРІШНЬО ПЕРЕМІЩЕНИХ ОСІБ В УКРАЇНІ: НАУКОВО-ПРАКТИЧНИЙ ОГЛЯД 140

ВИКОРИСТАННЯ ТЕКСТОВИХ РЕДАКТОРІВ В ПРОЦЕСІ НАВЧАННЯ ІНОЗЕМНИМ МОВАМ: СУЧАСНИЙ НАУКОВИЙ ОГЛЯД..... 148

ГЕНДЕРНІ АСПЕКТИ ПРОФЕСІЙНОГО СПІЛКУВАННЯ 153

УЖГОРОДСЬКА СТАНЦІЯ ЕКСТРЕНОЇ МЕДИЧНОЇ ДОПОМОГИ: ПОКАЗНИКИ ОСНОВНОЇ ДІЯЛЬНОСТІ..... 157

ВПЛИВ ВІЙНИ ПРОТИ РОСІЙСЬКОЇ ВОЄННОЇ АГРЕСІЇ НА ОТРИМАННЯ НАСЕЛЕННЯМ УКРАЇНИ СТОМАТОЛОГІЧНОЇ ДОПОМОГИ ПРИ ЗАХВОРЮВАННІ НА КАРІЄС 163

ЗАХВОРЮВАНІСТЬ НАСЕЛЕННЯ УКРАЇНИ НА ІНФЕКЦІЙНІ ЗАХВОРЮВАННЯ ПІД ВПЛИВОМ ВІЙНИ ПРОТИ РОСІЙСЬКОЇ ВОЄННОЇ АГРЕСІЇ 168

ВПЛИВ ВІЙНИ ПРОТИ РОСІЙСЬКОЇ ВІЙСЬКОВОЇ АГРЕСІЇ НА РІВЕНЬ ЗАБЕЗПЕЧЕННЯ ЛІКАРЯМИ-СТОМАТОЛОГАМИ СТОМАТОЛОГІЧНОЇ СЛУЖБИ СИСТЕМИ ОХОРОНИ ЗДОРОВ'Я МОЗ УКРАЇНИ 174

ВПЛИВ ВІЙНИ ПРОТИ РОСІЙСЬКОЇ ВІЙСЬКОВОЇ АГРЕСІЇ НА СТРУКТУРУ СТОМАТОЛОГІЧНОЇ СЛУЖБИ СИСТЕМИ ОХОРОНИ ЗДОРОВ'Я МОЗ УКРАЇНИ..... 180

ЗАХВОРЮВАНІСТЬ ЖІНОК ЗАКАРПАТСЬКОЇ ОБЛАСТІ НА РАК МОЛОЧНОЇ ЗАЛОЗИ В ПЕРЕДВОЄННИЙ ПЕРІОД..... 187

ВПЛИВ ВІЙНИ НА СТАН ГРОМАДСЬКОГО ЗДОРОВ'Я У РЕГІОНАХ УКРАЇНИ, ЯКІ ВІДДАЛЕНІ ВІД ЗОНИ АКТИВНИХ ВОЄННИХ ДІЙ ПРОТИ РОСІЙСЬКОЇ ВІЙСЬКОВОЇ АГРЕСІЇ 192

ОРГАНІЗАЦІЯ НАДАННЯ ОФТАЛЬМОЛОГІЧНОЇ ДОПОМОГИ ДИТЯЧОМУ НАСЕЛЕННЮ В КЛІНІЧНІЙ ЛІКАРНІ №8 МІСТА ДНІПРО 197

ВПЛИВ ВИМУШЕНОГО ПЕРЕСЕЛЕННЯ ВИКЛИКАНОГО ВІЙНОЮ ПРОТИ РОСІЙСЬКОЇ ВІЙСЬКОВОЇ АГРЕСІЇ НА ЯКІСТЬ ЖИТТЯ І ЗДОРОВ'Я ПЕРЕСЕЛЕНЦІВ..... 202

РОЛЬ МОЛОДІ В ТОТАЛІТАРНИХ ПРОЄКТАХ «OPERA NAZIONALE BALILLA» ТА «HITLERJUGEND» 208

БЕЗПЕКОВА ПОЛІТИКА КРАЇН РЕГІОНУ ЦЕНТРАЛЬНОЇ ТА СХІДНОЇ ЄВРОПИ В КОНТЕКСТІ РОСІЙСЬКОЇ АГРЕСІЇ 219

DIGITAL MARKETING STRATEGIES IN THE B2B SEGMENT: THE ROLE OF SEO PLATFORMS 229

ФІЗИЧНЕ ВИХОВАННЯ В ЗВО ТА ЗНАЧЕННЯ ФІЗИЧНОЇ ПІДГОТОВКИ ДЛЯ МАЙБУТНЬОГО ПРАЦІВНИКА ПРАВООХОРОННИХ ОРГАНІВ 241

АРХІТЕКТОНІКА НАСІННЄВИХ КУЩІВ ТА УРОЖАЙНІСТЬ НАСІННЯ РЕДЬКИ ЛОБО 251

ДОСЛІДЖЕННЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ РЕАГУВАННЯ ДСНС УКРАЇНИ НА НАДЗВИЧАЙНІ СИТУАЦІЇ З УРАХУВАННЯМ БОЙОВИХ ДІЙ 259

PARALLEL EXECUTION OF VERTEX, GEOMETRY, AND FRAGMENT SHADERS IN PARALLELIZED RENDERING PROCESSES 268

MANAGEMENT ASPECTS OF AGRICULTURAL PRODUCTION IN THE MODERN CONTEXT 274

ПРОБЛЕМАТИКА СУЧАСНИХ МІГРАЦІЙНИХ ПРОЦЕСІВ: СТАН ТЕОРЕТИЧНОГО ОПРАЦЮВАННЯ 283

КОНСТРУЮВАННЯ РЕКОМБІНАНТНОЇ ПЛАЗМІДИ IN SILICO ДЛЯ СИНТЕЗУ ФАКТОРА ЗГОРТАННЯ КРОВІ VIII 291

ІСТОРІЯ БОТАНІЧНИХ ДОСЛІДЖЕНЬ ASTRODAUCUS LITTORALIS (M. BIEB.) DRUDE НА ТЕРИТОРІЇ ПІВДНЯ УКРАЇНИ 302

Oleksandr Romanyuk
Doctor of Technical sciences, professor of the department of software engineering, Vinnytsia National Technical University,
Vinnytsia, Ukraine
https://orcid.org/0000-0002-2245-3364
Oleksii Bobko
PhD student, Vinnytsia National Technical University,
Vinnytsia, Ukraine
https://orcid.org/0009-0000-1753-279X

PARALLEL EXECUTION OF VERTEX, GEOMETRY, AND FRAGMENT SHADERS IN PARALLELIZED RENDERING PROCESSES

Abstract. The article examines methods and architectural approaches to the parallel execution of the three key software stages of the graphics pipeline – vertex, geometry, and fragment shaders. The levels of parallelism employed in modern GPU architecture are analyzed, along with the interaction between shaders, synchronization mechanisms, internal pipeline processing, and practical optimization strategies. An analysis of bottlenecks, load balancing approaches, and new mechanisms such as mesh shading is provided. The results may be useful for graphics engine developers, GPU architects, and researchers in the field of computer graphics.

Keywords: 3D rendering, GPU, parallelism, vertex shader, geometry shader, fragment shader, mesh shaders, Vulkan API, SIMD/SIMT, multi-GPU rendering.

Relevance and Novelty: In modern three-dimensional rendering, shaders – vertex, geometry, and fragment – are executed with an exceptionally high degree of parallelism on the graphics processing unit (GPU). The primary goal of parallelization is to ensure the efficient utilization of hardware resources, including arithmetic units,

caches, command buffers, and threads. To achieve this, various levels of parallelism are applied: pipelined, data parallelism, multithreading, and instruction-level parallelism. The vertex shader is responsible for computing the position of each vertex in space. The geometry shader operates on primitives, modifying or generating new ones. The fragment shader processes each pixel or fragment, forming the final appearance of the scene. These stages work in close interaction yet independently, which allows the GPU to overlap them in time to achieve maximum throughput.

Literature Review. The analysis of scientific and technical literature in the field of parallel execution of shaders in the graphics pipeline demonstrates a significant evolution of architectural approaches in modern GPUs. In[1], the performance of shaders on contemporary GPU architecture is thoroughly examined, with particular attention given to load distribution among shader units, the impact of pipeline structure, and catching features. The author provides a comparative analysis of vertex, geometry, and fragment shaders, which is directly relevant to the topic of parallel execution and the identification of bottlenecks.

The concept of pipeline design with flexible execution organization is presented in the Piko framework[2], which allows the explicit specification of the graphics pipeline structure while controlling parallelism at the task and tile levels. This approach emphasizes the need for precise control over resource allocation, which is critical for building efficient low-level API implementations. In combination with the Vulkan API, Piko demonstrates an approach to implementing multithreading in the context of GPU command queues, which is particularly valuable for parallel scene construction, shadow processing, lighting, and texturing.

New architectural mechanisms, particularly mesh shaders, are actively explored in documentation and technical reports. In[3], a clear comparison between the classical

[1] M. V. Moya, “Shader Performance Analysis on a Modern GPU Architecture,” Technical Report, 2020.
[2] A. Patney, S. Tzeng, K. Seitz, J. D. Owens, “Piko: A Design Framework for Programmable Graphics Pipelines,” ACM Transactions on Graphics, vol. 34, no. 4, 2015.
[3] GPUOpen, “From Vertex Shader to Mesh Shader,” AMD Developer Blog, 2021. [Online]. Available: https://gpuopen.com

shader model and mesh shading is presented, with an analysis of advantages in terms of throughput, visibility management, and redundancy elimination. Mesh shading combines the vertex and geometry stages into a single step, enabling efficient work with meshlet structures, which significantly improves data locality and scalability for complex scenes.

Practical examples of parallelism in rasterization and fragment processing are presented in technical forums, particularly in[4], where SIMD/SIMT mechanisms, warp/wavefront structure, tile organization, and caching in fragment shader execution are discussed. Although the source is non-academic, it provides accurate engineering observations on the implementation of parallel graphics at the driver and GPU ISA level. This is important for analyzing the fragment stage, which is the most resource consuming.

Special attention should be given to the work[5], which is devoted to the use of GPU shaders for visualization. In this study, the authors analyze not only architectural but also applied aspects, including image generation, buffer management, and the integration of lighting and post-processing. The work also points out the interconnections between pipeline stages and the importance of synchronization in their parallel execution.

Parallel rendering in multi-GPU environments is discussed in[6], where several approaches to frame partitioning across GPUs are proposed. The challenges of synchronization, load balancing, and intermediate data transfer are highlighted. This direction is particularly promising for large-scale rendering tasks or scientific visualization. In this case, the implementation of parallelism is not limited to a single graphics processor but extends to computing clusters.

[4] StackExchange, “Parallelism in GPU’s Rasterization Process,” Computer Graphics Stack Exchange, 2023. [Online]. Available: https://computergraphics.stackexchange.com

[5] M. Bailey, J. Zhang, and K. Moreland, “Using GPU Shaders for Visualization,” Journal of Computer Graphics Techniques, vol. 4, no. 2, pp. 1–16, 2015.

[6] B. Li, S. Chen, D. Huang, and Y. Wang, “Multi-GPU Parallel Pipeline Rendering with Frame Splitting,” IEEE Transactions on Visualization and Computer Graphics, vol. 27, no. 6, pp. 2678–2691, Jun. 2021.

Additionally, the generalized concepts of deferred shading mentioned in[7] demonstrate how parallel execution of shaders can be integrated into complex multi-stage lighting models. This approach makes it possible to optimize the rendering of complex scenes with many light sources, where the redistribution of computations across the stages of the graphics pipeline is of critical importance.

Thus, the literature analysis indicates the active development of multi-level approaches to parallelism in GPU graphics: from low-level shader organization and caching to high-level task management via APIs and multi-device rendering environments. The integration of technologies such as mesh shading, command queues, SIMD parallelism, and tile-based scheduling enables achieving new levels of efficiency and realism in three-dimensional rendering. The reviewed sources provide a foundation for further research in the areas of dynamic load balancing, hybrid CPU+GPU rendering, and AI-based shader scheduling.

Parallel Execution of Vertex, Geometry, and Fragment Shaders. Parallel execution of the vertex shader is based on the decoupling of processing each vertex. The scene is provided as an array of vertices, which are grouped into batches and passed to the GPU's computational units. Thanks to the SIMD/SIMT architecture, each shader processes a set of vertices simultaneously. For example, with 10,000 vertices, the GPU can process 32 vertices in parallel within each warp or wavefront. The geometry shader processes a single primitive – point, line, or triangle – and, if necessary, generates additional ones. Each instance of the geometry shader is launched independently. However, due to the smaller number of primitives compared to vertices or fragments, parallelism is limited here. Geometry shaders often exhibit lower performance and create a bottleneck, which is why modern engines attempt to minimize their use or replace them with more flexible solutions.

The fragment shader has the highest potential for parallel execution, since each fragment forming a pixel on the screen is processed independently. For a Full HD

[7] Wikipedia, "Deferred shading," [Online]. Available: https://en.wikipedia.org/wiki/Deferred_shading

image, the GPU simultaneously launches the processing of millions of fragments. These fragments are grouped by 2×2 or 4×4 blocks to optimize caching and reduce instruction divergence. Before processing, depth tests, stencil tests, and early-Z operations are applied to discard invisible fragments and reduce the workload. Fragment shaders are the most resource-intensive, as they often include computations for lighting, reflections, texturing, and post-processing.

Modern-generation GPUs employ a unified shader architecture – the unified shader model. This means that computational units are not rigidly assigned to a specific stage but can dynamically change their role. Thus, under excessive load on the fragment stage, some resources from the less loaded vertex stage can be reassigned. In addition, pipeline optimizations are used, allowing vertices from one part of the scene, geometry from another, and fragments from a third to be processed simultaneously. Such pipeline overlap minimizes idle time and ensures high throughput.

In the new NVIDIA architectures (Turing, Ampere) and AMD architectures (RDNA 2, 3), mesh shaders have been introduced – an innovative approach that combines the vertex and geometry stages into a single program. This allows processing not individual vertices but entire batches (meshlets), which helps to avoid duplication and improves visibility control. In addition, amplification shaders are employed, enabling selective activation of only those meshlets required for the current frame, thereby reducing the workload. Such mechanisms demonstrate significant efficiency improvements when working with massive scenes.

Parallelization is also supported at the API level. For example, in Vulkan, multiple command buffers can be used and executed in different queues – graphics, compute, and transfer. This enables the parallel execution of subtasks such as background texture updates, asynchronous shadow generation, or lighting. In frameworks like Piko, the user can explicitly define which tasks should be executed in parallel, how binning (scene partitioning into tiles) is performed, how computations are scheduled across tiles, and which queues are utilized.

Bottlenecks in the parallel execution of shaders include memory overload (especially in the fragment stage), buffer access conflicts, inefficient caching, excessive vertex duplication, and suboptimal geometry generation. To avoid these issues, several methods are applied: early depth testing, adaptive LOD, dynamic task scheduling, limiting the use of geometry shaders, and reducing excessive access to global memory. For example, using local buffers or registers decreases traffic to video memory, while regrouping fragment workloads within a single tile improves caching.

Scientific studies also consider the possibility of performing rendering on multiple GPUs simultaneously. In this case, the frame is divided into regions, each processed by a separate GPU, or different pipeline stages are distributed across GPUs – for example, one handling the vertex stage and another the fragment stage. Such an approach requires complex synchronization but can be beneficial for high-performance tasks or the visualization of large volumes of data.

Finally, promising directions for further research include automatic load balancing among shaders based on machine learning, heterogeneous rendering using CPU+GPU, adaptive shaders with priorities, the integration of mesh shading into WebGPU, and new models of shader scheduling in cloud environments. The application of these approaches will make it possible to extend the boundaries of image realism while maintaining performance.