УДК 539.3 **https://doi.org/10.36910/6775-2410-6208-2024-11(21)-13**

POROUS MATERIAL SOFTWARE ANALYSIS

ПРОГРАМНИЙ АНАЛІЗ ПОРИСТИХ МАТЕРІАЛІВ

Mikulich O.A., Doctor of Engineering, Professor (Lutsk National Technical University), Zaiakin D.K., Ph.D. student (Lutsk National Technical University)

Мікуліч О.А., д.т.н., проф. (Луцький національний технічний університет), Заякін Д.К., аспірант (Луцький національний технічний університет)

У роботі описані ключові моменти розробки та впровадження програмного забезпечення для аналізу пористості матеріалів. Перевагою використання розробленого продукту є можливість проведення експериментів на основі звичайних фотографій, які тепер можна отримати за допомогою звичайного телефону з достатньою точністю. Таке застосування значно розширює можливості проведення досліджень у виробництві без додаткових лабораторних експериментів.

Програма, що базується на мові програмування Swift та архітектурному шаблоні MVVM, є потужним інструментом для аналізу архітектурних об'єктів та матеріалів. Шаблон MVVM сприяє підтримуваності, знижуючи складність коду та покращуючи його читабельність, що дозволяє розробникам зосередитися на своїх конкретних областях експертизи. Використовуючи можливості продуктивності та оптимізації Swift, програмне забезпечення забезпечує плавний та зручний користувацький досвід, що особливо важливо для завдань обробки зображень. Розробка програмного забезпечення полегшується використанням IDE Xcode та емуляторів, які надають комплексні інструменти для створення, тестування та налагодження iOS-додатків. Крім того, програмне забезпечення використовує модульні тести для перевірки функціональності, підвищуючи якість та надійність коду.

Програма має інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам легко вибирати та аналізувати зони зображень, надаючи детальну інформацію про площу та пористість. Майбутні вдосконалення включатимуть розширені алгоритми для визначення щільності пор, безпечне зберігання, обмін та експорт результатів аналізу, що задовольняє різноманітні потреби досліджень. Цей додаток буде особливо корисним для архітекторів, інженерів та будівельників, спрощуючи процес проектування та забезпечуючи ефективне використання ресурсів. У цілому програмне забезпечення є універсальним та потужним інструментом для аналізу пористих матеріалів, а постійна розробка спрямована на подальшу оптимізацію та розширення можливостей.

The paper describes key moments in the development and implementation of software for analyzing the porosity of materials. The advantage of using the developed product is the ability to conduct experiments based on regular photographs, which can now be obtained with a regular phone with sufficient accuracy. Such an application significantly expands the possibilities of conducting research in production without additional laboratory experiments.

The program, based on the Swift programming language and the MVVM architectural pattern, is a powerful tool for analyzing architectural objects and materials. The MVVM pattern promotes maintainability by reducing code complexity and improving code readability, allowing developers to focus on their specific areas of expertise. By leveraging Swift's performance and optimization capabilities, the software ensures smooth and responsive user experiences, crucial for image processing tasks. The development of the software is facilitated by the use of Xcode IDE and simulators, which provide comprehensive tools for building, testing, and debugging iOS applications. Additionally, the software employs unit tests to verify functionality, enhancing code quality and reliability.

The program features an intuitive interface that allows users to easily select and analyze image zones, providing detailed information about the area and porosity. Future enhancements will include advanced algorithms for determining pore density, secure storage, sharing, and exporting of analysis results, catering to diverse research needs. This application will be particularly useful for architects, engineers, and builders, simplifying the design process and ensuring efficient use of resources. Overall, the software represents a versatile and powerful tool for analyzing porous materials, with ongoing development aimed at further optimization and expansion.

Ключові слова: пористі матеріали, програмне забезпечення, площа, аналіз, архітектура.

Keywords: porous materials, software, area, analysis, architecture.

Introduction. Porous materials have found significant application in modern industry due to their specific properties, allowing them to solve various important tasks in different fields, from construction to medicine. Experimental studies show [1] that one of the key aspects of this class of materials is determining their mechanical, physical, and other characteristics, such as porosity, and structural heterogeneity, which significantly affect the behavior of these materials during operation. It is not always possible to conduct experimental studies in laboratory conditions using special equipment, so this work is dedicated to the development of software that would allow determining the porosity of a material based on photographs of its structure.

Statement of Research Objectives and Tasks. The paper presents research on the capabilities of an early version of software aimed at solving the tasks described above. This software product has the potential for further development and expansion of its functions, which will improve the process of analyzing porous materials.

Main Part. The utilization of the MVVM (Model-View-ViewModel) architectural pattern (fig. 1) in the development of this software offers several advantages. In MVVM, the Model represents the data and business logic of the application, the View encompasses the user interface elements, and the ViewModel acts as an intermediary between the Model and the View, handling the presentation logic and exposing data to the View.

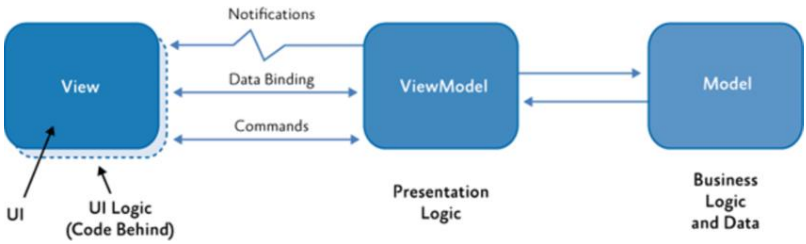

Figure 1 – MVVM pattern

By adhering to MVVM, the software achieves a high degree of separation of concerns, which facilitates modularization and promotes code reusability. This separation also simplifies testing, as each component can be tested independently, leading to more robust and reliable software.

Furthermore, the clear separation of responsibilities between the Model, View, and ViewModel enhances collaboration among developers working on different parts of the application. Developers can focus on their specific areas of expertise without interfering with the codebase of other components, leading to increased productivity and streamlined development processes.

The MVVM pattern promotes maintainability by reducing code complexity and improving code readability. With well-defined roles for each component, developers can easily understand the purpose and functionality of different parts of the software, making it easier to maintain and update the codebase over time.

By leveraging the MVVM architectural pattern, the software benefits from improved modularity, testability, collaboration, and maintainability, ultimately resulting in a more robust and efficient application development process.

The choice of Swift as the programming language for development brings additional benefits to the project. Swift's concise syntax and powerful features streamline development, allowing for faster implementation of complex functionalities. Its robust performance and optimization capabilities ensure smooth and responsive user experiences, particularly crucial for image processing tasks like those in this software. Moreover, Swift's active development community and extensive documentation provide ample resources and support for developers, facilitating rapid prototyping and iteration. In essence, Swift empowers developers to create innovative and reliable software solutions efficiently and effectively.

In the development process, certain program elements, such as text fields and buttons, were built upon existing components to leverage their functionality and ensure consistency across the application. These components were carefully separated into reusable modules, allowing them to be easily integrated and utilized throughout the entire software. This modular approach not only promotes code reusability but also simplifies maintenance and updates by isolating changes to individual components without impacting the overall application structure.

Furthermore, the software uses unit tests to verify the functionality of its various components and ensure they perform as expected. Unit tests are small, focused tests that validate individual units of code, such as functions or methods, in isolation. Their primary purpose is to detect and prevent regressions and bugs by automatically validating code behavior against expected outcomes. By implementing unit tests, developers can identify and address issues early in the development process, leading to higher code quality, improved reliability, and faster debugging. Additionally, unit tests serve as living documentation, providing insights into the intended behavior of code units and facilitating collaboration among development teams. Overall, the integration of unit tests enhances the robustness and maintainability of the software, ultimately contributing to its success in meeting user requirements and expectations.

The development of the software is facilitated by the use of Xcode IDE (Integrated Development Environment) and simulators, which play a crucial role in testing the application during its development lifecycle. Xcode, Apple's official IDE, provides a comprehensive suite of tools for building, testing, and debugging iOS applications. Its intuitive interface and powerful features streamline the development process, allowing developers to efficiently write, edit, and manage code.

One of the key features of Xcode is its built-in simulators, which enable developers to test their applications on virtual iOS devices without the need for physical hardware. These simulators accurately emulate the behavior and characteristics of various iOS devices, including different screen sizes and hardware configurations. This allows developers to verify the functionality and user experience of their applications across a wide range of devices, ensuring compatibility and consistency.

Additionally, Xcode provides robust debugging tools that help developers identify and resolve issues in their code quickly. Features such as breakpoints, variable inspection, and real-time code execution make it easier to diagnose and fix bugs, improving the overall quality of the software.

Overall, Xcode IDE and simulators are indispensable tools in the development of iOS applications, offering developers the resources and capabilities they need to create high-quality, reliable software efficiently.

The program has an intuitive interface that allows the user to easily select a zone on the image and change its size. The availability of such features allows for accurately determining the area of the selected zone and provides ease of use of the program. Let's consider the existing capabilities of the program. Selecting a zone on the image of porous aerated concrete (fig. 2) will allow determining its porosity and material properties.

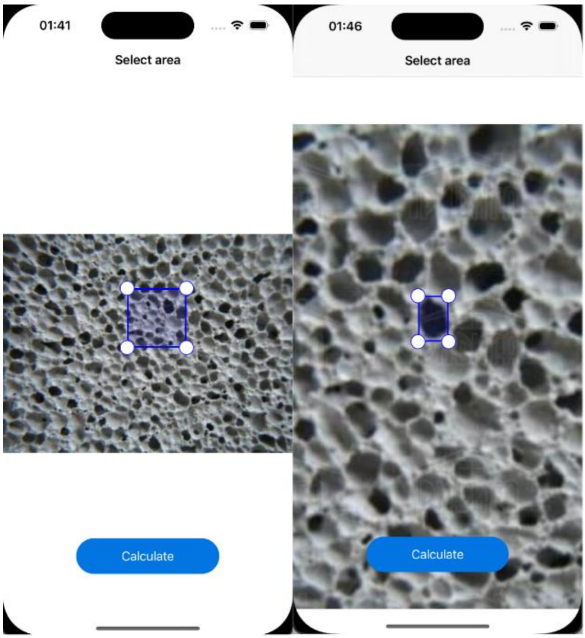

Figure 2 – Zone selection

The ability to change the size of the selected zone will allow the user to more accurately determine the area under investigation. The screen with the calculation of the size of the entire image will provide the user with a general overview and control over data processing. Using the screen with the calculation of the selected zone (fig. 3), the user can obtain detailed information about the area of the selected zone, which allows for a more accurate analysis of image areas.

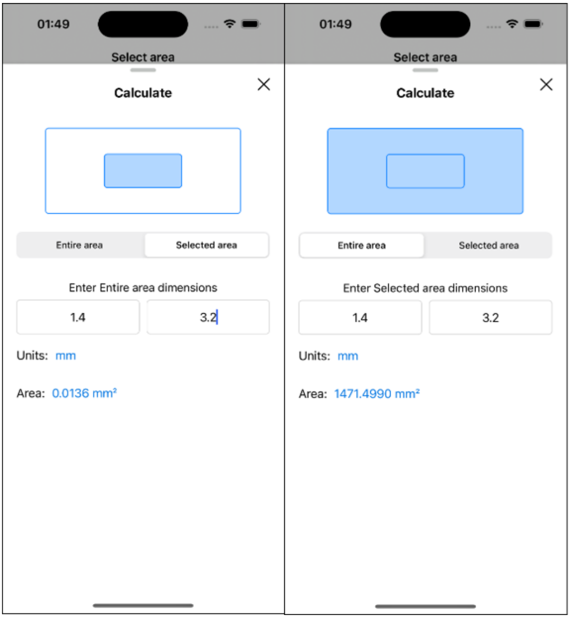

Figure 3 – Area calculation

Several approaches are known in the literature for determining the porosity of materials in laboratory conditions. One of them requires knowing the true density of the sample material and the density of the porous material itself [2]:

$$\Pi = \left(1 - \frac{\rho_v}{\rho_t}\right) \cdot 100\%, \quad (1)$$

where ρ_t — is the density of the solid material of the sample, $\rho_v = \frac{m}{V}$ — is the density of the porous material, m and V are the mass and volume of the sample with pores.

In addition, when determining porosity, the value of the relative volume fraction V_V [2]:

$$\Pi = (1 - V_V) \cdot 100\%, \quad (2)$$

where the relative volume fraction is defined as the ratio of the matrix volume to the total sample volume.

Both approaches allow determining the porosity of the material with sufficient accuracy in laboratory conditions using special equipment. To calculate the porosity of materials without using laboratory conditions, software has been developed that allows analysis based on photographs of the material structure. In the analysis of the structure of this software product, a combination of geometric and statistical approaches is applied in the calculation. The application considers several methods that allow for determining the size of pores in images.

In the current iteration, the application has begun implementing several techniques for analyzing the structure of materials. However, there are plans underway to optimize and expand these functionalities extensively. One such enhancement includes the integration of additional forms for selecting zones within the images, providing users with more flexibility and precision in their analyses.

Moreover, the software aims to incorporate advanced algorithms for determining pore density within the material structure. This enhancement will enable users to gain deeper insights into the distribution and characteristics of pores, further enriching the analytical capabilities of the application.

The development roadmap includes features for securely storing, sharing, and exporting analysis results, enhancing collaboration among researchers and professionals. Additionally, the application will offer capabilities for converting analysis results into various formats, catering to diverse research needs and preferences, thus ensuring flexibility and seamless integration with existing workflows and tools.

Overall, these planned optimizations and expansions signify the software's commitment to continuous improvement and innovation, aiming to provide users with a comprehensive and versatile tool for analyzing porous materials efficiently and effectively.

Conclusions. This application will be extremely useful for architects, engineers, and builders who regularly work with architectural objects and materials. Performing quick and accurate area calculations for the selected zone will simplify the design process and work with materials, ensuring efficient use of resources and high-quality results.

The developed program based on the Swift programming language and the MVVM architectural pattern is a powerful tool for analyzing architectural objects and materials. Its capabilities for determining the area within the selected zone along with porosity calculation methodologies make it an indispensable assistant for professionals in the field of architecture and construction. Further development and improvement of the application will take into account the needs of users and ensure its greater efficiency and usefulness.

References

1. Wypych, G.: Handbook of Foaming and Blowing Agents; ChemTec Publishing: Toronto, ON, Canada, 2007.
2. Carolis, S., Putignano, C., Soria L., Carbonen G.: Effect of porosity and pore size distribution on the elastic modulus of foams. *International Journal of Mechanical Sciences*, 2024. 261: 108661.