

Нижче описане ключове визначення розширеного критерію покриття діапазонами значень:

Визначення (range-sensitive def-use). Нехай $DU=(d(x,v), u(x,v'))$ – пара «визначення-використання». Позначимо через $C=\{C_1, C_2, \dots, C_m\}$ скінченне розбиття області значень змінної x на абстрактні діапазони. Тоді набір тестів T_{DU} задовольняє критерій range-sensitive def-use покриття відносно пари DU якщо для кожного діапазону $C_k \in C$ існує траса $t \in T_{DU}$, така що значення змінної x у вершині v' належить діапазону C_k .

Можливі різні сценарії впровадження запропонованого критерію: за наявності попередньо заданих діапазонів [4] можна спрямувати пошук безпосередньо на досягнення їх покриття (наприклад, методом перевірки моделі). Якщо діапазони заздалегідь не задані, можливе їх автоматичне генерування на льоту під час аналізу простору станів моделі. При цьому окремо виділяються критичні значення (наприклад, некоректність індексації, нуль в знаменнику).

На відміну від критерію all-uses, який орієнтується лише на структурне покриття пар «визначення-використання», запропонований метод додатково вимагає покриття семантично різних діапазонів значень, вдосконалюючи таким чином перевірку валідації вхідних даних. Метод дозволяє виявляти випадки, коли певні діапазони можливих значень залишаються неврахованими тестами, попри формальне досягнення покриття за класичними критеріями. Попередні дані застосування критерію свідчать про ~20% зростання кількості тестів у порівнянні з критерієм all-uses, при цьому спостерігається пропорційне зростання мутаційної оцінки.

1. Common Weakness Enumeration (CWE). 2024 CWE Top 25 Most Dangerous Software Weaknesses [Електронний ресурс]. – Режим доступу: https://cwe.mitre.org/top25/archive/2024/2024_key_list.html
2. Herrera, A., Payer, M., Hosking, A. DatAFLow: Toward a Data-Flow-Guided Fuzzer // ACM Transactions on Software Engineering and Methodology. – 2023. – Vol. 32, No. 5. – Article 132. – 31 p. DOI: 10.1145/3587156
3. Rapps, S., Weyuker, E. Data flow analysis techniques for test data selection // Proceedings of the International Conference on Software Engineering. – 1982. – P. 272–277.
4. Kolchin, A., Potiyenko, S. Extending Data Flow Coverage to Test Constraint Refinements // Lecture Notes in Computer Science, vol. 13274. – 2022. DOI: 10.1007/978-3-031-07727-2_17

КОНЦЕПЦІЯ СЕРВІСУ ОФЛАЙН МАП НА ОСНОВІ CESIUMJS

Ткаченко В.О.

Національний Університет “Кисво-Могилянська академія”

04655, м. Київ, вулиця Григорія Сковороди, 2, НаУКМА, Факультет інформатики (1 корп, кімн. 302).

E-mail: vo.tkachenko@ukma.edu.ua; телефон (063) 704 97 37

This paper presents a concept for enabling offline workflows with geospatial data in web applications built on CesiumJS. Since CesiumJS does not natively support offline use, the proposed service introduces a user-controlled process. The workflow begins with selecting a map region and associated datasets, which are preloaded into local persistent storage. During offline operation, rendering relies solely on cached imagery tiles and 3D Tiles, while user edits such as point additions, route creation, and object modifications are recorded in a structured JSON. Upon reconnection, the changes are transmitted to the backend and synchronized with the database, with conflict resolution applied if objects were modified in parallel. This cycle — preparation, offline work, and synchronization — ensures predictable behavior, preserves data integrity, and extends CesiumJS and 3D Tiles to environments where network connectivity is limited or unavailable.

У цій роботі запропоновано концепцію сервісу, що дає змогу працювати з геопросторовими даними в офлайн-режимі у веб-застосунках на основі CesiumJS. Підхід передбачає попереднє завантаження вибраного регіону мапи до локального сховища, збереження змін і об'єктів доданих під час роботи в режимі офлайн та

подальшу синхронізацію з базою даних після відновлення мережі. Розглянуто архітектурні рішення та процес роботи з даними — від підготовки плиток мапи і 3D-моделей до оброблення конфліктів під час зворотної інтеграції.

CesiumJS — це JavaScript бібліотека з відкритим кодом для візуалізації тривимірних геопросторових даних у веб-браузері.[1] Вона підтримує відображення рельєфу, супутникових знімків, векторних шарів та різних типів 3D-наборів, у тому числі у форматі 3D Tiles. Інструментарій бібліотеки включає API для розробки інтерактивних застосунків, а також навчальні приклади й документацію. У матеріалах проєкту підкреслюється потокова організація роботи з даними та можливість інтеграції з онлайн-сервісами зберігання і доставки, зокрема через платформу Cesium ion, що дозволяє працювати з великими наборами просторового контенту у веб-середовищі.

3D Tiles — відкритий стандарт (OGC Community Standard), призначений для ефективного пошарового стрімінгу та візуалізувати великі тривимірні геопросторові дані, такі як фотограмметрії, 3D-будівель, BIM/CAD, точкових хмар. Специфікація описує ієрархію плиток, формати плиток (b3dm, i3dm, pnts, cmpt) і механізми стилізації, що дозволяє клієнту завантажувати лише релевантні рівні деталізації залежно від масштабу та видимості. Завдяки цьому 3D-контент підвантажується поступово й доступний для інтерактивної роботи у веб-браузері.[2]

Важливо зазначити, що у стандартній конфігурації CesiumJS не має можливості працювати в офлайн-режимі. Усі приклади та документація бібліотеки передбачають доступ до онлайн-сервісів, зокрема до об'єктів рельєфу, плиток мапи та 3D Tiles, які зберігаються на серверах. Для багатьох сфер застосування це не є критичною проблемою, однак у випадках, коли користувач працює у польових умовах або в регіонах із відсутнім підключенням до Інтернету, виникає необхідність у локальному збереженні та подальшій синхронізації даних. Саме це мотивує до розробки сервісу який дозволяє працювати офлайн та дає можливість користувачу керувати процесом збереження та синхронізації.

Сесія починається з явної дії користувача — перемикання у режим підготовки даних. Інтерфейс дозволяє виділити територію і визначити набір потрібних шарів: рельєф, растрові зображення (WMTS), набори 3D Tiles тощо. Далі спрацьовує механізм асинхронного завантаження: обчислюються адреси плиток для всіх можливих масштабів, ініціюються паралельні запити з контролем черги, тайм-аутів і повторних спроб. На етапі валідації перевіряється мінімальна повнота покриття: відсутні елементи позначаються для довантаження або повідомляються користувачу.

Після успішного отримання ресурси зберігаються в локальне сховище. Для зображень це структуроване збереження за ключем схеми $\{z\}/\{x\}/\{y\}$; для 3D Tiles — кешування кореневого маніфесту набору (tileset.json) і піддерев плиток, включно з їхніми двійковими вмістами. Сховище організоване так, щоб забезпечити сталий доступ під час рендерингу (перезапуск браузера чи застосунку (PWA) не призводить до втрати даних). Для роботи з локальними файлами використовується File System API. Цей API дозволяє взаємодіяти з файлами на локальному пристрої користувача або на мережевій файлової системі, доступній користувачеві. Основна функціональність цього API включає читання файлів, запис або збереження файлів, а також доступ до структури каталогів.[3]

Після відключення мережі рендеринг працює винятково з локального джерела. Дані отримуються зі сховища за ключами плиток, 3D-набори відкриваються за локальними URL-посиланнями, а система візуалізації застосовує ті самі механізми рівнів деталізації, але без мережевих викликів. У цей час редакторські дії користувача — додавання точок інтересу, ліній/маршрутів, правки атрибутів об'єктів — не відправляються на сервер, а зберігаються у локальному JSON-файлі. Об'єкти мають ідентифікатор (новий або посилання на існуючий) та час події. Це дозволяє оновити об'єкти у безпечному порядку, коли мережа знову з'явиться.

Повернення в онлайн ініціюється користувачем. Об'єкти з JSON-файлу надсилаються через API. Якщо об'єкт змінювався також на сервері під час офлайн-періоду, спрацьовує політика розв'язання конфліктів: пріоритет за часовою міткою або

за явним вибором користувача. Після успішної інтеграції бекенд повертає остаточні ідентифікатори, локальні файли видаляються і клієнт переходить у звичайний мережевий режим.

Такий цикл дає можливість будувати передбачувану поведінку: користувач керує стартом та завершенням офлайн-сесії; регіон і набори даних завжди відомі; зміни не губляться; синхронізація відбувається чітко визначеними кроками. У веб-інтерфейсі відображається поточний статус та індикація завантаження під час отримання даних.

Розроблений підхід розширює можливості CesiumJS у напрямку автономної роботи, відкриваючи перспективи для застосування у сферах де з'єднання з мережею є слабким, або відсутнє повністю. Ключовим елементом підходу є керований користувачем цикл: вибір регіону карти та набору даних для попереднього завантаження, їх збереження у локальному файловому сховищі, накопичення змін під час офлайн-сесії та синхронізація з базою даних після відновлення доступу до мережі. Такий підхід забезпечує безперервність роботи, цілісність внесених змін і механізм їх інтеграції із серверною частиною. Модель дозволяє ефективно поєднати потокову природу CesiumJS із потребою зберігати дані локально.

Література:

1. CesiumJS Documentation — Learn & API. Доступ: cesium.com/learn/ (офіційні матеріали, API-довідник, приклади).
2. OGC 3D Tiles — Специфікація та огляд. Доступ: docs.ogc.org та репозиторій специфікації.
3. File System API - Concepts and Usage: Доступ: https://developer.mozilla.org/en-US/docs/Web/API/File_System_API (офіційні матеріали, API-довідник, приклади).

SYSTEM AND APPROACH FOR AUTOMATED PHOTO-LIKE FACIAL IMAGE GENERATION FOR USE IN TESTING, TRAINING OR EVALUATION

Bohdan Artiushenko, PhD (Candidate of Technical Sciences),

National University of Kyiv-Mohyla Academy

04655, Kyiv, Hryhorii Skovoroda St 2., Faculty of Informatics, Department of Informatics

E-mail: b.artiushenko@ukma.edu.ua, <https://orcid.org/0009-0007-6107-4205>

This research focuses on systems, apparatuses, and methods for automatically generating photo-like facial image items that may be used on an exam or test. The test may be used for evaluating a test-taker's proficiency in familiar faces recognition, training in facial recognition, prosopagnosia or other similar purpose or goal. A combination of generative artificial intelligence, image and face recognition and manual check is proposed and evaluated.

Prosopagnosia, or face blindness, is widely considered as hardly improbable cognitive disorder, thus training on face recognition is often suggested [2]. It's also widely believed that it's difficult for normal humans to distinguish faces of another nationality.