



МОБІЛЬНА КІБЕРФІЗИЧНА СИСТЕМА ДЛЯ ДИНАМІЧНОГО ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ ПРО ОБ'ЄКТИ НА ЦИФРОВІЙ КАРТІ МІСЦЕВОСТІ

З кожним роком реальний і віртуальний світи стають все ближчі один до одного, утворюючи технічну базу кіберфізичних систем. Кіберфізична система об'єднує кібернетичний та фізичний простори, інтегруючи обчислювальні та фізичні процеси за допомогою давачів і виконавчих пристроїв. Одним із видів кіберфізичних систем, у яких компоненти системи здатні динамічно змінювати своє місцезнаходження, є мобільні кіберфізичні системи, а обчислювальною платформою для їх реалізації слугують персональні мобільні пристрої. Розв'язано науково-практичну задачу щодо розроблення математичного та програмного забезпечення для динамічного відображення інформації про об'єкти на цифровій карті місцевості; проаналізовано підходи до візуалізації об'єктів на цифровій карті місцевості та подано принципи візуалізації даних із використанням технології доповненої реальності; запропоновано метод для динамічного відображення інформації про об'єкти реальної світу на цифровій карті місцевості, через призму камери персонального мобільного пристрою (ПМП), що використовує дані із картографічних сервісів та давачів ПМП, відображаючи інформацію із використанням технології доповненої реальності; розроблено архітектуру та реалізовано мобільну кіберфізичну систему (МКФС) для динамічного відображення інформації про об'єкти на цифровій карті місцевості для ПМП на базі операційної системи Android із використанням мови програмування Java, Android API та Google Maps API.

Ключові слова: доповнена реальність; картографічна система; персональний мобільний пристрій; Google Maps API; Android.

Вступ. Мобільні кіберфізичні системи (англ. mobile cyber-physical system, MCPS) є одним із видів кіберфізичних систем, у яких компоненти системи здатні динамічно змінювати своє місцезнаходження (White, et al., 2011). Стрімкий розвиток ринку ПМП, що характеризується зростанням потужностей процесорів, збільшенням часу автономної роботи, широкою комунікаційною периферією, набором вбудованих давачів і тісною інтеграцією з інтернет-сервісами дає змогу використовувати їх у ролі обчислювальних платформ для реалізації МКФС. Перелічені переваги ПМП дають змогу використовувати їх для моніторингу та контролю роботи фізичних об'єктів.

До основних чинників, що вплинули на розвиток МКФС, відносять (Bales, et al., 2012):

- зростання кількості та зниження вартості ПМП;
- покращення технічних характеристик та функціоналу ПМП, які нічим не поступаються персональним комп'ютерам за істотно менших розмірів та ваги;
- упровадження нових технологій безпроводного зв'язку (3G, Mobile WiMAX, EDGE, Bluetooth);
- розвиток мобільних операційних систем та платформ для ПМП (Android, iPhone OS, Windows Mobile, BlackBerry);
- розвиток високорівневих мов та технологій програмування ПМП (Java, Qt, Windows Phone SDK, Objective-C, Android SDK);
- можливість розповсюдження власних мобільних додатків через App Store постачальників ПМП та проведення експериментів з великою кількістю учасників по всьому світі;

- комплектація ПМП потужними вбудованими давачами;
- розвиток мобільних хмарних технологій та обчислень.

Постановка завдання дослідження. Сьогодні відбувається стрімкий розвиток геосервісів, які надають змогу користувачу зорієнтуватись у незнайомому місці, відшукати потрібний об'єкт, прокласти маршрут, відобразити координати місцезнаходження користувача ПМП на цифровій карті місцевості. Особливо популярними ці сервіси є серед туристів, більшість з яких подорожуючи по світу використовують персональні мобільні помічники, які в режимі реального часу надають всю необхідну туристу інформацію.

За результатами проведеного аналізу програмних продуктів для динамічного відображення інформації про об'єкти на цифровій карті місцевості з'ясовано, що всі вони працюють із 2D картами, які є аналогами паперових карт місцевості (Shevchuk, Kohut & Boiko, 2014). Одним із недоліків такого типу карт є неможливість вирішення проблеми "топографічного кретинізму". Під "топографічним кретинізмом" розуміють абсолютне невміння людини орієнтуватися на місцевості, навіть із наявністю карти, що часто супроводжується відчуттям страху заблукати у незнайомій місцевості (White, et al., 2011). Величезні мегаполіси з розгалуженими вулицями, складними дорожніми розв'язками й архітектурно однаковими будинками ускладнюють їм життя.

Тому постає задача створення МКФС для динамічного відображення інформації про об'єкти на цифровій карті місцевості, в основу якої буде закладено принци-

Цитування за ДСТУ: Шевчук Р. П., Мельник А. М. Мобільна кіберфізична система для динамічного відображення інформації про об'єкти на цифровій карті місцевості. Науковий вісник НЛТУ України. 2017. Вип. 27(4). С. 170–176.

Citation APA: Shevchuk, R. P., & Melnyk, A. M. (2017). Mobile Cyber-Physical System for Dynamic Representation of Information About Objects on Digital Maps. Scientific Bulletin of UNFU, 27(4), 170–176. <https://doi.org/10.15421/40270436>

ного світу на цифровій карті місцевості, в основі якого лежить процедура знаходження кутових відстаней до об'єктів з використанням циліндричної системи координат. Цей метод охоплює такі етапи:

- знаходження об'єктів на цифровій карті місцевості;
- визначення магнітного азимуту;
- побудова векторів від ПМП до об'єктів на цифровій карті;
- обчислення кута між вектором напрямку ПМП та вектором об'єкта на карті;
- побудова піктограми об'єкта на екрані ПМП із використанням технологій доповненої реальності та виведення інформації про нього на екран ПМП.

На рис. 3-6 наведено ілюстрацію етапів роботи запропонованого методу.

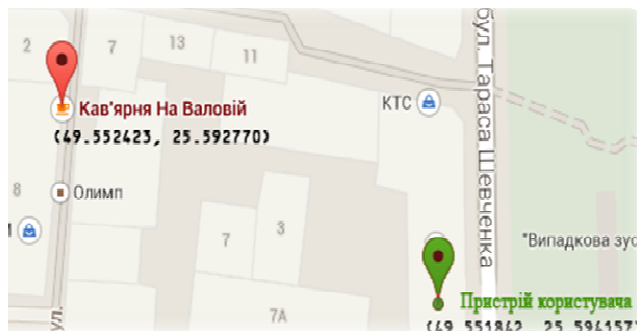


Рис. 3. Знаходження об'єктів на цифровій карті



Рис. 4. Визначення магнітного азимуту

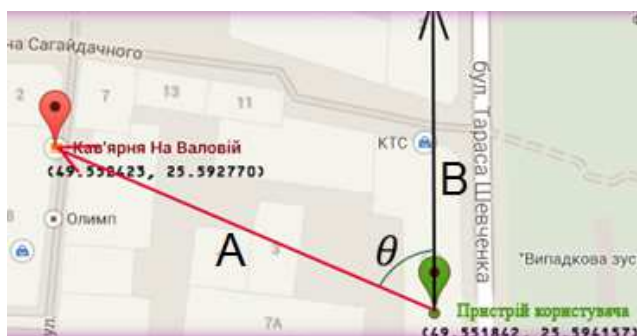


Рис. 5. Побудова векторів від ПМП до об'єктів на карті

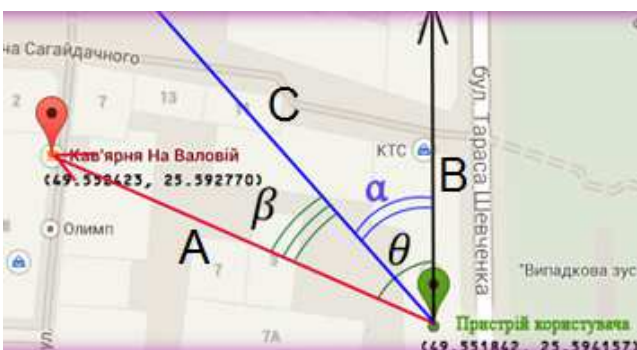


Рис. 6. Обчислення кута між вектором напрямку ПМП та вектором об'єкта на карті

Побудова векторів від ПМП до об'єктів на карті та обчислення кута між вектором напрямку ПМП та вектором об'єкта на карті виконується із використанням таких формул:

$$\theta = \arccos\left(\frac{(AB)}{|A| \times |B|}\right), \quad (1)$$

де: θ – кут від пристрою до нульового вектора; A – вектор від точки положення пристрою до точки місця; B – вектор відліку;

$$\alpha = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \quad (2)$$

де: α – кут зору пристрою; A – вектор від точки положення пристрою до точки об'єкта; B – вектор відліку;

$$\beta = \alpha - \theta, \quad (3)$$

де β – кутова відстань до пристрою.

Орієнтація у просторі здійснюється на основі колекції координат, отриманих із Інтернет-сервісу Google Maps. Процес завантаження колекції координат складається з таких етапів:

- визначення місця знаходження користувача ПМП;
- передача поточних координат на веб-сервер Google;
- отримання колекції координат навколишніх об'єктів.

Використання циліндричної системи координат є ефективнішим, тому що для збереження інформації про об'єкт, необхідно зберігати тільки значення кута, на який воно відхиляється від нульового вектора – напрямку на північ. Така ж сама інформація присутня і для самого пристрою, що значно спрощує сам процес орієнтації. Додатковою інформацією є також відстань до об'єкта.

Особливістю запропонованого методу є використання циліндричної системи координат під час роботи із геомагнітним сенсором ПМП, що дає змогу зменшити обчислювальну складність базового алгоритму програмної системи.

Особливості розроблення та експлуатації МКФС.

На рис. 7, 8 подано діаграми компонентів модулів розробленої МКФС.

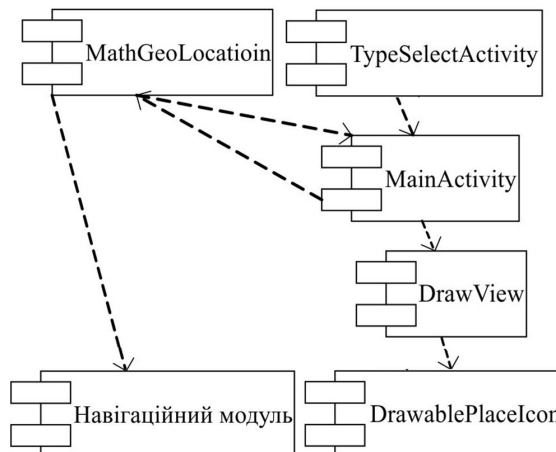


Рис. 7. Діаграма компонентів модуля геоінформаційного моніторингу

Користувачський інтерфейс виступає своєрідним мостом між віртуальним світом, який формується системою, реальним світом та користувачем. Завдяки його роботі запускаються процеси орієнтації пристрою у просторі та отримання списку місць реального світу навколо користувача. Також за посередництва графічного інтерфейсу відбувається взаємодія між навігаційним модулем та модулем геоінформаційного моніторингу. Коли надходить інформація про вибір певного

типу місця, модуль геоінформаційного моніторингу посилає запит навігаційному модулю на отримання списку найближчих місць, той, своєю чергою, з'єднується з сервісами Google та надсилає їм необхідні дані. Після

чого модуль геоінформаційного моніторингу відображає результати своєї роботи за допомогою графічного інтерфейсу.

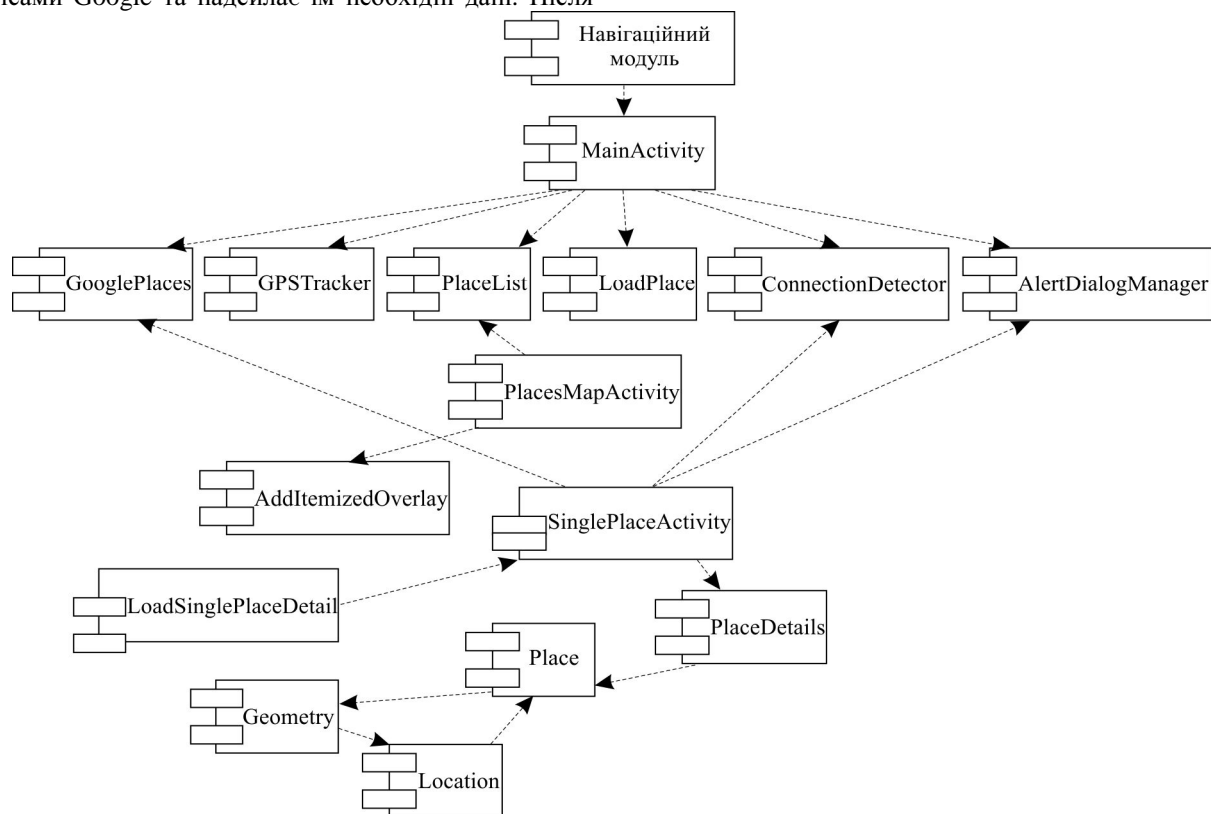


Рис. 8. Діаграма компонентів навігаційного модуля

На рис. 9 подано діаграму розміщення, яка відображає фізичні взаємозв'язки між програмними та апаратними компонентами системи.

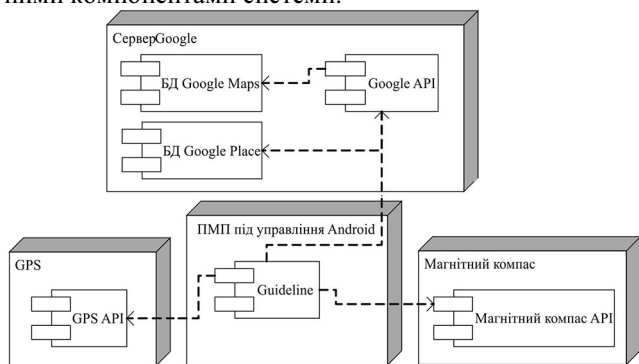


Рис. 9. Діаграма розміщень

Для графічного представлення поведінки модулів програмної системи побудовано їх діаграми діяльності (рис. 10, 11).

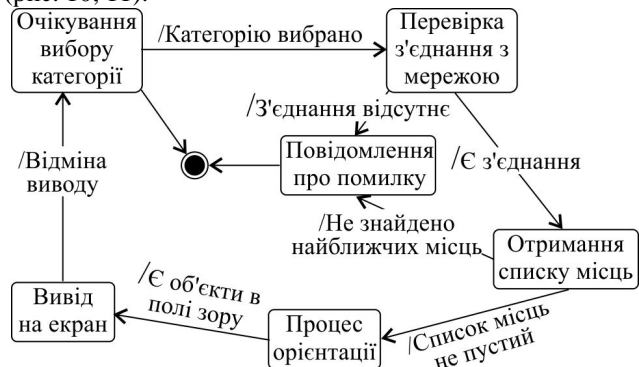


Рис. 10. Діаграма діяльності модуля геоінформаційного моніторингу

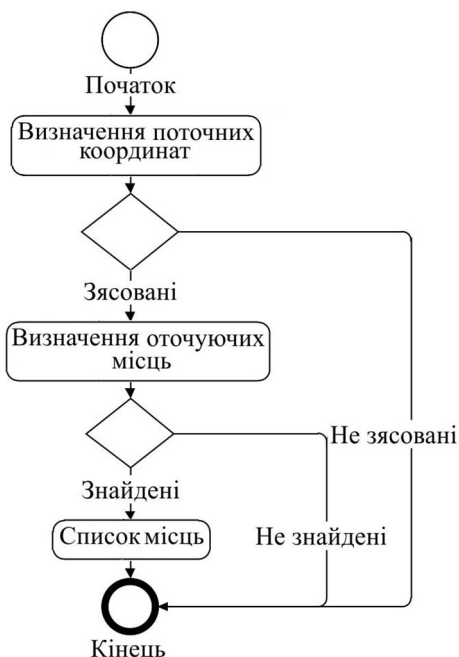


Рис. 11. Діаграма діяльності навігаційного модуля

На них, у вигляді графа, представлено дії системи, та переходи між ними, які є або результатами попередніх дій, або ж їх викликають.

Процедуру оброблення даних із геомагнітного давача персонального мобільного пристрою здійснює модуль геоінформаційного моніторингу. Основною його задачею є визначення кутової відстані від точки розміщення пристрою до об'єктів, розміщених на карті. Цей модуль розрахований на роботу з навігаційним модулем, з якого він отримує необхідну інформацію, а саме

географічні координати пристрою та координати визначних місць на карті.

Класом, без якого неможливо було б здійснювати коректні розрахунки положень об'єктів, є клас точки, значеннями координат якого можуть бути числа з плаваючою комою, тобто типу `double`. Оскільки у стандартних пакетах мови Java такий клас був відсутній, його було створено у проєкті.

Класом, який виконує основну функцію модуля, а саме визначення градусних відстаней до об'єктів, є клас `MathGeoLocation`. Клас виконує перетворення координат об'єкта з декартової системи координат у циліндричну, обрахунок відхилення у градусах, і пізніше розрахунок розміщення піктограми об'єкта на дисплеї пристрою, якщо об'єкт потрапляє в кут зору камери.

Методи цього класу є статичними, тобто їх виклик не потребує створення екземпляру класу.

Вихідний код класу `Point` наведено нижче:

```
package com.guideline;
public class Point {
    public double x;
    public double y;
    public Point(double X, double Y)
    {
        this.x=X;
        this.y=Y;
    }
}
```

Першим етапом алгоритму є перетворення координат об'єктів із декартової системи у циліндричну. Для цього, після отримання списку об'єктів, вони передаються в метод `transformPlacesPoints`. Нижче наведено сигнатуру методу.

```
static public List<PlacePoint> transformPlacesPoints(List<Place> places, Point myPoint)
```

Цей метод на вхід отримує колекцію об'єктів та географічні координати пристрою. Використовуючи ці дані, спочатку обчислюється кут між напрямом до об'єкта і віссю абсцис.

```
double A = Math.abs(myPoint.x - placePoint.x);
double B = Math.abs(myPoint.y - placePoint.y);
double degrees = Math.toDegrees(Math.atan(B/A)).
```

Після цього визначається, в якій четверті відносно точки розміщення пристрою знаходиться об'єкт та обчислюється значення азимута об'єкта.

```
if(myPoint.x > placePoint.x && myPoint.y < placePoint.y){
    degrees = 270 + (90 - degrees);
}
else if(myPoint.x > placePoint.x && myPoint.y > placePoint.y){
    degrees = 180 + (90 - degrees);
}
else if(myPoint.x < placePoint.x && myPoint.y > placePoint.y){
    degrees = 90 + degrees;
}
else if(myPoint.x < placePoint.x && myPoint.y < placePoint.y){
    degrees = 90 - degrees;
}
```

Ці дані зберігаються у список, і повертаються в викликаючий метод.

Перед викликом методу для визначення кутової відстані до об'єкта, дані, зчитані з геомагнітного давача ПМП, передаються у клас `KalmanFilter`, який виконує фільтрацію цих даних.

Для початку роботи з фільтром його потрібно проініціалізувати. Конструктор приймає на вхід 4 параметри, а саме: Q – значення шуму вимірювання; R – значення шуму обладнання; F – відношення попереднього

реального значення до поточного; H – відношення вимірюваного значення до реального.

Після цього, при кожному знятті показів з давача, їх необхідно передати в метод `Correct`. Код цього методу наведено нижче:

```
public void Correct(double data)
{
    //time update - prediction
    this.X0 = this.F*this.State;
    this.P0 = this.F*this.Covariance*this.F + this.Q;
    // measurement update - correction
    double K = this.H*this.P0/(this.H*this.P0*this.H+this.R);
    this.PrevState = this.State;
    this.State = this.X0 + K * (data - this.H * this.X0);
    this.Covariance = (1 - K * this.H) * this.P0;
}
```

Після цього, для визначення кутової відстані до об'єкта, з основного потоку викликається метод `getOnScreenLocation`. Він повертає значення, яке відповідає розміщенню піктограми місця на екрані пристрою. Нижче можна побачити його сигнатуру.

```
getOnScreenLocation(PlacePoint placePoint, int screenWidth, int viewAngle, double azimuth)
```

Як видно із сигнатури методу, він приймає 4 параметри, а саме:

- `placePoint` – координати об'єкта;
- `screenWidth` – ширину дисплея в пікселях;
- `viewAngle` – кут зору камери;
- `azimuth` – азимут пристрою.

Тут наведено код цього методу:

```
public static int getOnScreenLocation(PlacePoint placePoint, int screenWidth, int viewAngle, double azimuth)
{
    double pixelsInDegree = screenWidth/viewAngle;
    double onScreenLocation = 0;
    double degrees = getDegreeDeviation(placePoint, azimuth);
    if(degrees < 0 && degrees > -(viewAngle/2))
        onScreenLocation = (screenWidth / 2) + Math.abs(degrees) * pixelsInDegree;
    else if(degrees < viewAngle / 2)
        onScreenLocation = (screenWidth / 2) - degrees * pixelsInDegree;
    return onScreenLocation!=0.0? (int) onScreenLocation: -1;
}
```

Метод, який розраховує градусні відстані, є метод `getDegreeDeviation`. Нижче представлено сигнатуру методу:

```
static private double getDegreeDeviation(PlacePoint placePoint, double azimuth)
```

Як видно із сигнатури, метод приймає два вхідні параметри: `placePoint` – координата об'єкта, `azimuth` – азимут пристрою. Метод повертає значення кута типу `double` – число з плаваючою комою.

Для відображення піктограм на дисплеї створено два класи: `DrawView` та `DrawablePlaceIcon`. Клас `DrawablePlaceIcon` описує собою комплексну графічну компоненту для відображення інформації про місце. `DrawView` виконує малювання піктограм місць на дисплеї. Він зберігає колекцію об'єктів типу `DrawablePlaceIcon` і за допомогою методу `onDraw` виконує рисування піктограм.

Якщо об'єкт потрапляє в поле зору камери, його піктограма рисується на дисплеї пристрою у відповідній позиції. Екранну форму з нанесеними піктограмами зображено на рис. 12.

Для забезпечення функціонування модуля потрібно спочатку, отримавши список географічних об'єктів, діз-

натися чи входить він в поле зору камери і відповідно до цих даних створити новий список об'єктів піктограм.



Рис. 12. Екранна форма з піктограмами об'єктів

Систему розроблено у програмному середовищі Eclipse ADT на мові програмування Java з використанням Android SDK.

Для коректної роботи системи не потрібно встановлювати додаткові програми чи бібліотеки. Для стабільного функціонування МКФС, потрібне постійне під'єднання до мережі Інтернет.

У розробленій системі реалізовано такі функції:

- вибір категорії об'єктів;
- виведення на екран знайдених об'єктів;
- виведення списку знайдених об'єктів;
- виведення детальної інформації про об'єкт.

Запустивши застосунок, користувач потрапляє в головне меню, на якому пропонують вибрати категорію об'єктів, які він хотів би побачити на екрані свого ПМП (рис. 13).

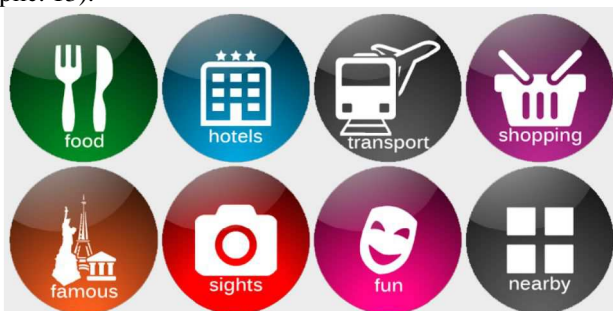


Рис. 13. Головне меню системи

Після вибору користувачем певної категорії, тримаючи ПМП перед собою в горизонтальному положенні, через призму камери буде відображатися місцевість із навколишніми об'єктами відповідно до вибраної категорії. Список знайдених об'єктів завантажується із бази даних Google Maps після визначення поточного місця користувача.

Тримаючи ПМП перед собою, але перевернувши його з горизонтального у вертикальне положення, на екрані з'явиться список усіх знайдених об'єктів вибраної користувачем категорії. Зробивши клік по одному із знайдених об'єктів у вертикальному положенні ПМП, відкриється його детальний опис. Екранну форму детального опису об'єкта можна побачити на рис. 14.

У роботі проведено тестування продуктивності розробленої МКФС.

Висновки. У роботі розв'язано науково-практичну задачу щодо розроблення математичного та програмного забезпечення для динамічного відображення інформації про об'єкти на цифровій карті місцевості. При цьому отримано такі результати:

1. Проаналізовано підходи до візуалізації об'єктів на цифровій карті місцевості. Подано принципи візуалізації

даних із використанням технології доповненої реальності.

2. Запропоновано метод для динамічного відображення інформації про об'єкти реального світу на цифровій карті місцевості, через призму камери ПМП, який, на відміну від відомих, використовує циліндричну систему координат під час роботи із геомагнітним сенсором ПМП, що дало змогу зменшити обчислювальну складність.
3. Розроблено архітектуру мобільної кіберфізичної системи для динамічного відображення інформації про об'єкти на цифровій карті місцевості та спроектовано модулі системи.
4. На етапі програмної реалізації, завдяки мові програмування Java у середовищі програмування Eclipse ADT із використанням Android API та Google Maps API, програмно реалізовано МКФС з динамічним відображенням об'єктів на цифровій карті місцевості для ОС Android.



Рис. 14. Детальна інформація про знайдений об'єкт

Перелік використаних джерел

- Azuma, R. A. (2004). Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 3(5), 355–385.
- Bales, E., Nikzad, N., Quick, N., Ziftci, C., Patrick, K., & Griswold, W. (2012). *Citisense: Mobile Air Quality Sensing for Individuals and Communities*. In *Pervasive Health*, 260 p.
- Choudhury, T. et al. (2008). The Mobile Sensing Platform: An Embedded System for Activity Recognition. *IEEE Pervasive Comp.*, 7(2), 32–41.
- Froehlich, J., Dillahunt, T., Klasnja, P., Manko, J., Consolvo, S., Harrison, B., & Landay, J. (2009). UbiGreen: Investigating a Mobile Tool for Tracking and Supporting Green Transportation Habits. In *Proceedings of the 27th international conference on Human factors in computing systems* (pp. 1043–1052). ACM SenSys, Berkeley, CA.
- Gay, V. C., & Leijdekkers, P. A. (2007). Health Monitoring System Using Smart Phones and Wearable Sensors. *International Journal of Assistive Robotics and Mechatronics*, 8(2), 29–36.
- Marino, A., Caccavale, F., Parker, L., & Antonelli, G. (2009, June). Fuzzy Behavioral Control for Multi-Robot Border Patrol. In *Proceedings of the 17th Mediterranean Conference on Control and Automation, Thessaloniki, Greece*, (pp. 25–29). Raleigh, NC, USA, Nov.
- Marino, A., Parker, L., Antonelli, G., Caccavale, F., & Chiaverini, S. (2009). A Modular and Fault-Tolerant Approach to Multi-Robot Perimeter Patrol. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, (pp. 135–138). Guilin, China, December.
- Melnik, A. (2016). Cyber-Physical Systems Multilayer Platform and Research Framework. *Advances in Cyber-Physical Systems*, 1(1–6), 1–9.
- Melnik, A. O. (2014). Kiberfizychni systemy: problemy stvorennia ta napriamy rozvytku. *Visnyk Natsionalnoho universytetu "Lvivska politekhnika". Series: Kompiuterni systemy ta merezhi*, 806, 154–161. [in Ukrainian].
- Miluzzo, E., Wang, T., & Campbell, A. T. (2010). EyePhone: Activating Mobile Phones With Your Eyes. *To appear in Proc. of The Second ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds (MobiHeld'10)* (pp. 230–238). New Delhi, India, August 30

- Miluzzo, E., Lane, N. D., Fodor, K., Peterson, R. A., Lu, H., Musolesi, M., & Shane, B. (2008). Eisenman, Xiao Zheng, Andrew T. Campbell, Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application. In *Proceedings of 6th ACM Conference on Embedded Networked Sensor Systems (SenSys '08)* (pp. 5–7). Raleigh, NC, USA, Nov.
- Mobile Millennium. (n.d.). UC Berkeley/Nokia/NAVTEQ. Retrieved from: <http://traffic.berkeley.edu/>
- Saponas, T., Lester, J., Froehlich, J., Fogarty, J., & Landay, J. (2008). *iLearn on the iPhone: Real-Time Human Activity Classification on Commodity Mobile Phones*. University of Washington CSE Tech Report UW-CSE-08-04-02, 368 p.
- Shevchuk, R. P. (2015). Pryntsyropy pobudovy ta funktsionuvannya mobilnykh kiberfizychnykh system. Kiberfizychni systemy: dosiahnennia ta vyklyky: mater. pershoho nauk. seminaru (pp. 90–95). Lviv: NVF "Ukrainski tekhnologii". [in Ukrainian].
- Shevchuk, R. P., Kohut, A. V., & Boiko, Ya. V. (2014). Personalnyi mobilnyi pomichnyk turysta z dynamichnym vidobrazhenniam ob'ektiv na tsyfrovii karti mistsevoosti. *Suchasni kompiuterni informatsiini tekhnologii: mater. IV Vseukrainskoi shkoly-seminaru molodykh vchenykh i studentiv* (pp. 167–169). Ternopil: FO-P V. B. Shpak. [in Ukrainian].
- Thiagarajan, A., Ravindranath, L., LaCurts, K., Toledo, S., Eriksson, J., Madden, S., & Balakrishnan, H. (2009, November). VTrack: Accurate, Energy-Aware Road Traffic Delay Estimation Using Mobile Phones. In *Proceedings of the 14th international conference on Human factors in computing systems* (pp. 942–950). ACM SenSys, Berkeley, CA.
- Thompson, C., White, J., Dougherty, B., & Schmidt, D. (2009). Optimizing Mobile Application Performance with Model-Driven Engineering. In *Proceedings of the 7th IFIP Workshop on Software Technologies for Future Embedded and Ubiquitous Systems* (pp. 104–109). ACM SenSys, Berkeley, CA.
- White, Ju., Clarke, S., Dougherty, B., Thompson, C., & Schmidt, D. (2011). R&D Challenges and Solutions for Mobile Cyber-Physical Applications and Supporting Internet Services. *Springer Journal of Internet Services and Applications*, 2(21), 134–139.

Р. П. Шевчук, А. Н. Мельник

Тернопольский национальный экономический университет, г. Тернополь, Украина

МОБИЛЬНАЯ КИБЕРФИЗИЧЕСКАЯ СИСТЕМА ДЛЯ ДИНАМИЧЕСКОГО ОТОБРАЖЕНИЯ ИНФОРМАЦИИ ОБ ОБЪЕКТАХ НА ЦИФРОВОЙ КАРТЕ МЕСТНОСТИ

С каждым годом реальный и виртуальный миры становятся все ближе друг к другу, образуя техническую базу киберфизических систем. Киберфизическая система объединяет кибернетическое и физическое пространства, интегрируя вычислительные и физические процессы с помощью датчиков и исполнительных устройств. Одним из видов киберфизических систем, в которых компоненты системы способны динамически изменять свое местоположение, являются мобильные киберфизические системы, а вычислительной платформой для их реализации служат персональные мобильные устройства.

Решена научно-практическая задача по разработке математического и программного обеспечения для динамического отображении информации об объектах на цифровой карте местности; проведен анализ подходов к визуализации объектов на цифровой карте местности и представлены принципы визуализации данных с использованием технологии дополненной реальности; предложен метод для динамического отображения информации об объектах реального мира на цифровой карте местности, через призму камеры персонального мобильного устройства (ПМУ), использующий данные с картографических сервисов и датчиков ПМУ, отображая информацию с использованием технологии дополненной реальности; разработана архитектура и реализована мобильная киберфизическая система (МКФС) для динамического отображения информации об объектах на цифровой карте местности для ПМУ на базе операционной системы Android с использованием языка программирования Java, Android API и Google Maps API.

Ключевые слова: дополненная реальность; картографическая система; персональное мобильное устройство; Google Maps API, Android.

R. P. Shevchuk, A. M. Melnyk

Ternopil National Economic University, Ternopil, Ukraine

MOBILE CYBER-PHYSICAL SYSTEM FOR DYNAMIC REPRESENTATION OF INFORMATION ABOUT OBJECTS ON DIGITAL MAPS

Cyber-physical systems (CPS) refer to a promising class of systems featuring intimate coupling between the "cyber" intelligence and the "physical" world. Cyber-physical systems are advanced engineering systems where computing and communication are carefully designed to achieve intimate integration with the physical dynamics. One of the types of cyber-physical systems where system components are able to dynamically change your location is a mobile CPS. Computing platforms for implementing mobile CPS are personal mobile devices. Mobile CPS integrates distributed sensing with computing and ubiquitous connectivity of the Internet. Mobile CPS also coordinates computational, virtual, and physical resources and facilitates the interaction of digital world with physical world, potentially driving the pervasive effect in the citizens' everyday life anytime and anywhere. Thus, mobile CPS could provide a convenient and economical platform that facilitates sophisticated and ubiquitous intelligent applications between humans and the surrounding physical world. The paper solved a scientific and practical task to develop mathematical and software for dynamic representation of information about objects on digital maps; analysis approaches for visualization objects on a digital map and presented principles of data visualization using augmented reality technology. The authors proposed the method for dynamic representation of information about objects in the real world on digital map, through the prism of a personal mobile device (PMD) camera, which uses information from the sensors PMP and mapping services and representation of information using augmented reality technology. We also proposed the architecture and implemented a mobile cyber-physical system for dynamic representation of information about objects on the digital map areas for PMD an Android-based operating system, programming language Java, Android API and Google Maps API.

Keywords: augmented reality; mapping system; personal mobile device; Google Maps API; Android.

Інформація про авторів:

Шевчук Руслан Петрович, канд. техн. наук, доцент, Тернопільський національний економічний університет, м. Тернопіль, Україна. Email: rulezz.sh@gmail.com

Мельник Андрій Миколайович, канд. техн. наук, доцент Тернопільський національний економічний університет, м. Тернопіль, Україна. Email: melnyk.andriy@gmail.com