

Отже, аналіз наведених результатів показує, що використання мереж Моделі Геометричних Перетворень до ущільнення зображень дає покращені результати (співвідношення *якість/час навчання*) порівняно з існуючими методами, причому результати компресії чорно-білих та кольорових зображень виявилися співвимірними між собою при однакових початкових умовах.

### Висновки

1. Розроблений метод, що ґрунтується на засадах навчання нейроподібних структур МГП з використанням представлення чисел з фіксованою комою, забезпечує ефективне ущільнення даних, зокрема без втрат інформації і може розглядатися як подальший крок у розвитку парадигми МГП.

2. Застосування формату чисел з фіксованою комою полегшує апаратну реалізацію нейромереж МГП високої продуктивності і швидкодії.

1. Сэломон Д. Сжатие данных, изображений и звука. – М.: Техносфера, 2004. 2. Osovski S. Sieci neuronowe dla przetwarzania informacji. Oficyna wydawnicza Politechniki Warszawskiej. Warszawa 2000. 3. Christopher M. Bishop. Neural Networks for Pattern Recognition. Oxford University Press (1999). 4. Kramer M. A. Autoassociative neural networks. Massachusetts inst. technology, dep. chemical eng., lab. intelligent systems process eng., Cambridge MA 02139, ETATS-UNIS. 5. A. Jerzov, A. Shumsky. Neurocomputing and its application in economic and business. MIFI 1998. 6. Ткаченко Р.О., Дорошенко А.В. Вдосконалення нейромережних методів класифікації в завданнях інтелектуального аналізу даних за допомогою методу імітації відпалу металу // Вісник Нац. ун-ту “Львівська політехніка”: Комп’ютерні системи проектування. Теорія і практика. – 2007. – № 591. – С. 33–37. 7. Tkachenko R., Tkachenko P., Tkachenko O., Schmitz-Lenders J. Geometrical transformation machine // Computer sciences and information technologies: Materials of the international conference CSIT”2007. – Lviv, 2007. – Pp. 52–53. 8. Ткаченко Р.О., Дорошенко А.В. База моделей на основі моделі геометричних перетворень для систем підтримки прийняття рішень // Комп’ютерні технології друкарства. – 2007. – № 17. – С. 21–28.

УДК 004.942

М. Медиковський, О. Шуневич

Національний університет “Львівська політехніка”,  
кафедра автоматизованих систем управління

## МЕТОД ЗАСТОСУВАННЯ СТАНДАРТУ OPC ДЛЯ СИНТЕЗУ СИСТЕМ УПРАВЛІННЯ

© Медиковський М., Шуневич О., 2010

**Розглянуто способи взаємодії вбудованих систем з програмним забезпеченням комп’ютера, встановлено переваги застосування стандарту OPC, проаналізовано способи взаємодії OPC-клієнтів із OPC-серверами. Досліджено OPC-сервери і здійснено системний аналіз можливостей щодо симуляції отримання/передачі даних.**

**Some ways of interactions between embedded systems and software of PC, defined benefits of using OPC-standard, analyzed the ways of interactions between OPC-servers and OPC-clients. Investigated OPC-servers and implemented the system analysis of simulation mode for a receiving/sending data.**

### Вступ

Більшість автоматизованих систем управління будуються на базі промислових контролерів. Контролери регулюють технологічні параметри, аварійні сигнали; вони є засобом захисту і блокування (нижній рівень системи). Оброблена контролерами інформація передається до комп’ю-

теризованих систем, які є робочим місцем оператора-технолога, де відбувається подальше оброблення даних і представлення оператору в інтуїтивно зрозумілому вигляді цих даних (верхній рівень автоматизованої системи управління технологічним процесом – АСУ ТП). У цьому випадку одним із перспективних засобів реалізації таких процесів є технологія OLE (Object Linking and Embedding – зв’язування і вбудовування об’єктів), запроваджена компанією Microsoft для інтеграції (взаємодії) між різними офісними програмами. Під інтеграцією розуміють використання об’єктів однієї програми в іншій. Починаючи з версії OLE 2.0, в її основу було покладено компонентну модель COM (*Component Object Model*) [1].

Використання OLE для аналізу і синтезу систем управління передбачає побудову, дослідження і застосування технології OPC (OPC – це абревіатура від *OLE for process Control*). Тут і постає актуальним розроблення ефективного методу застосування індустріального стандарту OPC, який дає змогу під’єднати програмний продукт до пристроїв або систем різних виробників, використовуючи єдиний програмний інтерфейс.

### Аналіз існуючих методів розв’язання задачі

Оскільки більшість контролерів працюють під управлінням вбудованої операційної системи, то слова «контролер» і «вбудована система» у цій статті мають однакове значення.

Практичний досвід та аналіз публікацій [2] дають можливість виділити три способи взаємодії вбудованих систем з інструментальним і цільовим програмним забезпеченням комп’ютера:

– «Одна вбудована система – одна програма». Кожній вбудованій системі відповідає своє унікальне програмне забезпечення (рис. 1). У більшості випадків його розробляє фірма - розробник вбудованої системи. Використовуючи цю методику, можна добитися оптимальних результатів в плані швидкості обміну інформацією, одержати можливість використання переваг кожної вбудованої системи. З іншого боку – підтримка і модернізації програмного забезпечення для великої кількості вбудованих систем потребують додаткових ресурсів;

– «Одна вбудована система – багато програм». Кожна система надає відкритий для всіх або для певних фірм-розробників ПЗ (програмного забезпечення) інтерфейс, і будь-яка програма, що підтримує його, може взаємодіяти із вбудованою системою (рис. 2). У цьому випадку розроблення і виробництво систем полегшується, оскільки створити програмне забезпечення може як розробник вбудованої системи, так і інші особи;

– «Багато вбудованих систем – багато програм». Кожна вбудована система має свій інтерфейс (програмний). Вона надається разом з драйвером, який забезпечує стандартний інтерфейс, однаковий для всіх вбудованих систем (драйвер перетворює внутрішній програмний інтерфейс на зрозумілий для всіх програмістів). Користувачські системи також надають стандартний інтерфейс, тому кожна вбудована система може взаємодіяти з будь-якою з них (рис. 3).

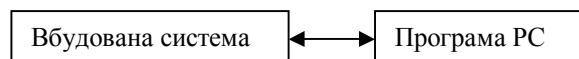


Рис. 1. Вбудована система з унікальним програмним забезпеченням



Рис. 2. Вбудована система з відкритим інтерфейсом для програмного забезпечення

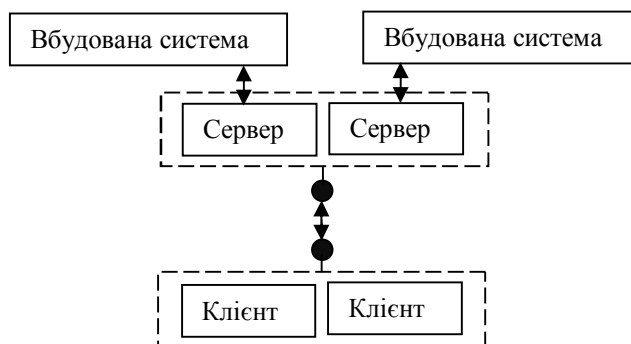


Рис. 3. Вбудована система з стандартним інтерфейсом для програмного забезпечення

Реалізація кожного із цих способів взаємодії за індивідуальними алгоритмами та інструментарієм породжує певні труднощі при експлуатації таких систем, тому логічно в цьому випадку застосувати стандарт OPC, який забезпечує програмному продукту доступ до пристроїв або систем різних виробників, використовуючи єдиний програмний інтерфейс.

### Доступні інструменти

OPC – це сімейство програмних технологій, які надають єдиний інтерфейс для управління об'єктами автоматизації і технологічними процесами.

Ця технологія визначає два класи програм:

- 1) OPC-сервер, який безпосередньо взаємодіє з апаратурою;
- 2) OPC-клієнт, який отримує дані від OPC-сервера для подальшого оброблення, а також передає до нього команди управління.

Отже, використовуючи специфікацію OPC, виробник апаратних засобів має можливість розробити програму-сервер, яка буде забезпечувати доступ до даних програмам-клієнтам різних виробників програмного забезпечення. Своєю чергою, виробники ПЗ мають можливість одержувати дані для оброблення від декількох різних систем за стандартним інтерфейсом [3].

Структурну схему взаємодії між апаратними засобами, серверними і клієнтськими програмами показано на рис. 4:

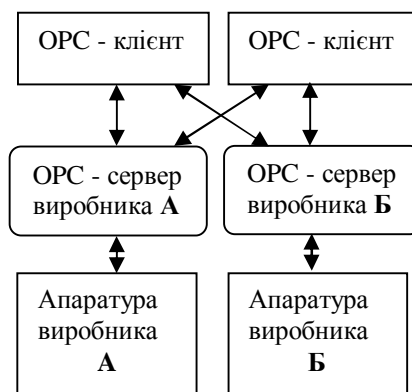


Рис. 4 Взаємодія апаратури, OPC-серверів і OPC-клієнтів

Отже, OPC-сервер – це програма або бібліотека, яка виконується на комп'ютері, до якого безпосередньо під'єднано вбудовану систему. Клієнт – це програма, якій потрібний доступ до вбудованої системи. В кожного сервера може бути декілька клієнтів, а клієнт може обмінюватися даними з декількома серверами.

Загалом система стандарту OPC містить головні складові:

- **DataAccess** – основний стандарт, який описує набір функцій для обміну даними. Цей стандарт забезпечує безпосередній доступ до даних реального часу і за його допомогою здійснюються управляючі дії (програмування на C++);

- Alarm&Event – надає функції повідомлення про різні події: аварійні ситуації, інформаційні повідомлення та ін.;
- HistoricalData – доступ до раніше збережених даних, тобто доступ до архіву даних [4].

Всі OPC-сервери реалізують як мінімум стандарт DataAccess. Технологія OPC містить як обов'язкові для реалізації інтерфейси, так і необов'язкові. В кожному інтерфейсі також є обов'язкові і необов'язкові функції. Відповідно до моделі СОМ, набір функцій кожного інтерфейсу є фіксований. Необов'язковість функцій означає, що при виклику вони повертають код помилки, який говорить про те, що функція не реалізована. Наприклад, кожна одиниця даних DataAccess сервера ідентифікується унікальним іменем. У мінімальному об'ємі реалізації сервер не надає інформації про імена своїх одиниць даних. У ширшій реалізації є спеціальний інтерфейс для отримання списку імен одиниць даних і інших властивостей даних (тип, підтримка тимчасових міток тощо). В мінімальній реалізації клієнт може отримати доступ до даних OPC-сервера, тільки якщо він знає найменування даних із інших джерел (наприклад, якщо є документація для OPC - сервера).

### Дослідження способів під'єднання OPC-клієнта до OPC-сервера

Раніше вже було зазначено, що кожний OPC-сервер є СОМ сервером, зв'язок з яким здійснюється через OPC інтерфейси з використанням технології Microsoft DCOM. Всі OPC-сервери мають унікальний ідентифікатор CLSID. Клієнту необхідно знати його, щоб під'єднатися до сервера.

Технологія СОМ передбачає три основні типи серверів: Локальний(local), Віддалений(remote) і в вигляді DLL(inproc):

- Локальний сервер означає, що виконавчий модуль буде завантажуватися на цій самій машині, що і клієнт. Принцип роботи такий самий, як в inproc, але з єдиною відмінністю – inproc DLL завантажується в адресний простір клієнта, тоді як local завантажується окремим процесом.

- Віддалений сервер означає, що виконавчий модуль буде завантажуватися на віддаленій по мережі машині.

Проаналізувавши більшість способів, за допомогою яких можна отримати дані від OPC-сервера, необхідно виділити такі:

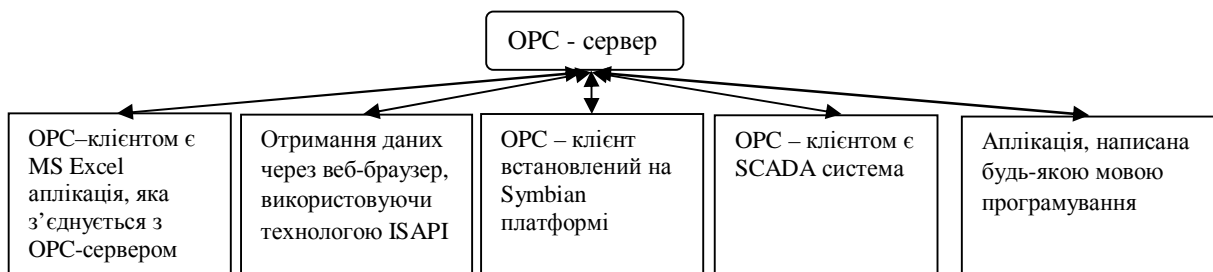


Рис. 5. Основні способи отримання/передачі даних від/до OPC-сервера

**Рішення на основі MS Excel.** Щоб створити програмне забезпечення для АСУ ТП на невеликих підприємствах, в наукових і конструкторських закладах, зручно використовувати дешеві засоби розроблення, такі як Microsoft Excel, який є практично на кожному персональному комп'ютері. Цей підхід може здатися дивним, адже в жодній документації з Excel не міститься інформація про таку можливість. Тим не менше, Excel можна використовувати для вирішення багатьох завдань промислової автоматизації. Особливо зручно застосовувати MS Excel для завдань автоматизації експериментальних досліджень, коли алгоритми розв'язання задачі потрібно змінювати під час експерименту і оперативно виконувати різні перетворення над одержаними даними [5].

В принципі, мова Visual Basic for Application (VBA), яка входить до складу MS Excel, дає змогу розв'язати практично будь-яку задачу, що стоїть перед розробником SCADA - системи. Щодо під'єднання апаратної частини, то слід сказати, що MS Excel підтримує такі технології: 1) динамічного обміну даними (DDE); 2) динамічно зв'язуваних бібліотек (DLL); 3) модель багатоконпонентних об'єктів (COM).

Отже, MS Excel може бути успішно використаний для розв'язання задач автоматизації технологічних процесів і лабораторних досліджень. Його особливістю є широкі можливості щодо оброблення даних, інтуїтивно зрозумілі графічні засоби, зручний і добре відпрацьований інтерфейс. Головним недоліком такого підходу є те, що MS Excel не призначений безпосередньо для розв'язання задач, які стоять перед SCADA системами.

**Взаємодія з OPC-сервером через Internet.** Необхідно зазначити, що одним з основних недоліків технології COM є непристосованість для роботи в глобальних мережах. Разом з тим, передача даних на великі відстані дуже цікавить користувачів АСУ ТП, оскільки можна здійснювати віддалений моніторинг.

У цьому випадку, щоб отримати дані від OPC-сервера, найчастіше використовують технологію ISAPI (Internet Server Application Programming Interface). Технологія ISAPI нагадує за своєю суттю технологію CGI, в якій web-сервер для генерації частини сторінки викликає спеціальні програми. В ISAPI замість спеціальних програм використовується DLL. Отже, можна створити HTML сторінку, яка буде відображати значення змінних [6].

Недоліком такої взаємодії є те, що всі операції з оброблення даних, їх візуалізації, інтелектуалізації покладено на серверні скрипти, швидкість роботи яких, своєю чергою, залежатиме від кількості web-клієнтів.

Сьогодні OPC Foundation розробляє специфікацію – OPC XML. Її мета – розробити гнучкий і зручний інтерфейс для обміну даними через OPC, використовуючи XML (Extensible Markup Language) в аплікаціях Internet/Intranet. Функції XML дають змогу зручно записувати будь-які структури даних і водночас передавати дані у вигляді XML-файлів, які є зручними для пересилання через Internet [7].

**Взаємодія з OPC-сервером через мобільний телефон.** Компанія Warice пропонує OPC-клієнта для платформи Symbian. Цей програмний продукт надає можливість використовувати мобільний телефон як OPC-клієнт. Наразі продукт є сумісним з декількома комунікаторами фірми Nokia [8].

**SCADA-система.** Одним із компонентів АСУ ТП, який здатний надавати персоналу технологічну інформацію, є SCADA-система. Саме SCADA-система є найпоширенішим OPC-клієнтом, який розв'язує такі типові задачі: диспетчерський моніторинг і збирання даних про технологічний процес; управління за наявності чітких алгоритмів і повної формалізованої моделі об'єкта управління.

**Створення OPC-клієнта різними мовами програмування.** Слід одразу сказати, що OPC-клієнта можна створити лише тими мовами, які підтримують технологію COM. Для написання аплікації програмісту потрібна тільки взаємодія з OPC-специфікаціями. З погляду програмування існують декілька мов для написання клієнтської програми: C/C++, C#, Visual Basic, Delphi і т.д. Сьогодні більшість OPC-клієнтів розробляються мовою C/C++.

#### Результати досліджень

Вище обґрунтовано основні способи побудови OPC-клієнтів. Процес розроблення OPC-клієнта вимагає використання OPC-сервера, який є проміжною ланкою між апаратною і клієнтською частиною. У більшості випадків розробники не мають доступу до вбудованих систем (тобто не володіють апаратурою, яка безпосередньо надає дані для оброблення) і їм доводиться обирати серед великого числа OPC-серверів такі, які якомога точніше відповідають технічним вимогам. Крім того, реальною є ситуація, коли відсутня інформація про доступні режими та параметри того чи іншого OPC-сервера. Якщо ж інформація доступна, то, як правило, у неповному вигляді. До великого числа серверів не подається жодної документації щодо алгоритмів опрацювання і параметрів даних. Також важко зробити висновок про наявність тих чи інших додаткових функцій сервера.

Тому актуальним є експериментальні дослідження відомих серверів з метою встановлення їх можливостей. При цьому ефективно реалізувати такі дослідження у межах технічних вимог можна тільки для серверів із функцією симуляції режимів отримання/передачі даних.

Для розв'язання такої задачі досліджено OPC-сервери, які рекомендовані засновниками стандарту OPC, і здійснено системний аналіз можливостей щодо симуляції отримання/передачі даних.

При цьому враховано, що основними характеристиками одиниць даних є: тип, номінальні значення, допустимі межі, критичні межі, можливий алгоритм зміни значення, період оновлення значення, достовірність елемента даних.

Метою дослідження було:

- визначити, які типи даних підтримує сервер;
- ознайомитися з всіма доступними видами алгоритмів симуляції;
- довідатись, чи є можливість налаштувати ієрархію одиниць даних;
- інформація про доступність вихідного коду є надзвичайно важливою;
- визначити, які OPC-стандарти підтримуються цим сервером;
- встановити, чи є можливість імпорту/експорту одиниць даних, тобто чи є наявна зовнішня база даних, в якій можна зберігати інформацію про ці елементи даних.

Алгоритм дослідження передбачав використання Prosys OPC-client як OPC-клієнта. За чергою встановлювалися сервери Prosys OPC, Northern Dynamic, Lectus ModBus, Gray Simulator, Icons OPC Simulator і досліджувалися їх параметри. Спочатку сервери перевірялися на наявність основних характеристик: доступні типи даних, наявні режими симуляції, відкритість коду; потім визначалися додаткові характеристики, такі як: можливість налаштування ієрархії одиниць даних, підтримувані OPC-стандарти, наявність зовнішньої бази даних для збереження інформації про одиниці даних та ін.

Результати досліджень характеристик OPC-серверів наведено у таблиці. Цю таблицю можна застосувати як шаблон для вибору OPC-сервера при розв'язанні різних задач.

### Характеристика OPC-серверів

Характеристики	Сервери				
	2				
1					
<b>Наявність графічного інтерфейсу</b>	+	+	+	-	+
<b>Доступні типи даних</b>					
Наявність логічного типу даних (Logical)	+	+	+	+	+
Наявність цілочисельного типу даних (Integer)	+	+	+	+	+
Наявність дійсного типу даних (Float, Real)	+	-	+	+	+
Наявність текстового типу даних (Text)	+	+	+	-	+
Наявність типу даних дата (Date)	-	-	+	-	+
Наявність типу даних масив (Array)	+	+	-	-	-
<b>Алгоритми симуляції</b>					
Static/Memory (статичне значення)	+	+	+	+	+
Random чисел на встановленому діапазоні	+	-	+	-	-
Random чисел без діапазону	-	-	-	+	-
Random символів/слів	-	-	-	+	-
Random дати	-	-	-	+	-
Random числового ряду	-	-	-	+	-
Random значення з встановленого списку	-	-	+	-	+
SawTooth/Step/Raw/Ramp/Increment/Decrement (ступінчаста зміна значення)	+	+	+	+	+
Sinusoid (зміна значень за синусоїдою)	+	-	-	+	+
Triangle (зміна значень за трикутником)	+	-	-	-	+
Square (значення в квадраті)	-	-	-	+	+
<b>Встановлення умов безпеки</b>	+	-	-	-	+
<b>Налаштування ієрархії</b>	-	-	+	-	+
<b>Створення нових змінних</b>	-	-	+	+	+
<b>Відкритість коду</b>	+	+	-	+	-

1	2				
<b>Стандарт</b>					
DA 1.0	-	+	-	+	+
DA 2.0	-	+	-	-	-
DA 2.05a	+	+	+	+	+
DA 3.0	-	+	-	+	-
HAD 1.2	-	-	+	-	-
XML-DA	-	-	-	-	+
<b>Період опитування кожного елемента значень</b>	-	-	+	-	+
<b>Наявність бази даних одиниць даних</b>	-	-	-	-	+
	<b>Prosys OPC</b>	<b>Northern Dynamic</b>	<b>Lectus ModBus</b>	<b>Gray Simulator</b>	<b>Icons OPC Simulator</b>

Крім цього, встановлено такі важливі особливості серверів:

**Prosys OPC.** Особливим у цьому сервері є те, що можна встановлювати «верхнє небезпечне» і «нижнє небезпечне значення» (low alarm limit=20, high alarm limit=80, low warning limit=30, high warning limit=70). Також сервер має можливість виводити список клієнтів, які в цей момент отримують дані від сервера.

**Lectus ModBus.** Особливою функцією цього OPC-сервера є можливість розраховувати значення елемента даних за заданою формулою. Також є можливість створювати власні вузли (ієрархію); встановлювати період опитування кожного елемента даних.

**Gray Simulator.** Основним недоліком цього OPC-сервера є те, що він не має власного графічного інтерфейсу і те, що немає можливості налаштовувати елементи даних. Під'єднуватись до нього можна з клієнта, використовуючи CLSID або ProgID ідентифікатори. Також цей сервер є безплатним. Особливістю цього сервера симуляції є наявний текстовий алгоритм генерації, під час якого здійснюється симуляція отримання кольорів, днів тижня, генерація серійних номерів, чисел буквами і ін. Генерація (поточна або випадкова) дати/часу також є важливою перевагою цього сервера над іншими.

**Icons OPC Simulator.** Перевага цього OPC-сервера полягає в тому, що користувач має можливість створити базу даних, в якій зберігатимуться дані про теги (елементи даних): ім'я тега, діапазон існування, налаштування тривоги і ін. Також даний симулятор вирізняється з поміж інших тим, що користувач має можливість розробляти математичні формули, за якими надалі оброблятимуться елементи даних. Сервер зорієнтований на роботу із вбудованими системами.

### Висновки

На підставі системного аналізу методів взаємодії вбудованих систем з програмами комп'ютера встановлено переваги застосування OPC-стандарту над класичними підходами (можливість налаштування сервера в режимі симуляції, доступність стандарту та ін).

Шляхом експериментальних досліджень визначено основні характеристики та параметри OPC-серверів, що забезпечує можливість застосовувати сформований шаблон при синтезі систем управління.

1. *Стандарт OPC путь к интеграции разнородных систем [Електронний ресурс]. – Режим доступу: <http://www.asutp.ru/?p=600094>* 2. *Ковязин Р.Р., Платунов А.Е. Применение технологии OPC. – Санкт-Петербургский государственный университет информационных технологий, меха-*

ники и оптики: Стаття, обзор, 2003. 3. Технология OPC. OPC-сервер и OPC-клиент [Электронный ресурс]. – Режим доступа: <http://www.novosoft.by/Ency/opc.htm> 4. Применение стандарта OPC [Электронный ресурс]. – Режим доступа: <http://www.pikprogress.ru/opc.php>. 5. Денисенко В.В. Построение автоматизированных испытательных стендов с помощью системы RealLab и MS Excel // Компоненты и технологии, 2001. – №6. – С. 96–98 6. Григорьев А.Б. Взаимодействие с OPC-сервером через Internet [Электронный ресурс]. – Режим доступа: <http://asutp.ru/?p=600450> 7. Технология OPC. Основные принципы и преимущества (Руководство пользователя ИСБ Интеллект, 2007. [Электронный ресурс] – Режим доступа: [http://www.itv.ru/download/documentation/intellect/html/index.html?int\\_scada.htm](http://www.itv.ru/download/documentation/intellect/html/index.html?int_scada.htm) 8. OPC programmers' connection [Электронный ресурс]. – Режим доступа: <http://opcconnect.com/tooltech.php#embedded>.

УДК 004.932.4

Ю. Шийка, Р. Шувар

Львівський національний університет  
імені Івана Франка

## НЕПРОПОРЦІЙНЕ МАСШТАБУВАННЯ ЗОБРАЖЕНЬ МЕТОДОМ SEAM CARVING І ВИБІР ЕНЕРГЕТИЧНИХ ФУНКЦІЙ ДЛЯ НЬОГО

© Шийка Ю., Шувар Р., 2010

**Розглянуто методи масштабування растрових зображень зі зміною співвідношення сторін. Досліджено метод Seam Carving непропорційного масштабування. Запропоновано методи побудови енергетичних функцій для нього, порівняно результати їх використання.**

**Methods of raster image resizing with change of the aspect ratio are described. Seam Carving method of content aware image resizing is researched. Different methods of calculation of energy functions for Seam Carving are offered, comparison of results of resizing using different energy functions is made.**

### Постановка проблеми

Одна з найчастіших задач обробки растрових зображень – масштабування. Це зумовлено тим, що в сучасних умовах різноманітності цифрових пристроїв з різними типами дисплеїв різних розмірів і з різними співвідношеннями сторін постають задачі представлення графічних даних зі зміною співвідношення сторін.

Багато стандартів, як, наприклад HTML, підтримують метод динамічного представлення інформації: розміщення таблиць, тексту тощо. Однак щодо зображень, то вони далеко не такі гнучкі і в кращому випадку можуть набувати певних наперед визначених розмірів або змінюватись тільки пропорційно. За необхідності відобразити зображення на дисплеї, чи надрукувати його на папері певного розміру і в ряді інших задач часто виникає необхідність масштабування зображення зі зміною співвідношення сторін. Ця необхідність визначає актуальність дослідження методів непропорційного масштабування.

### Аналіз існуючих методів

Стандартні методи зміни співвідношення сторін зображення полягають у непропорційному масштабуванні (scaling) або кадрванні (crop) – вирізанні потрібної частини зображення. Методи масштабування використовують передискретизацію зображення для уникнення втрати дрібних