МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Дніпропетровський національний університет
імені Олеся Гончара

ISSN 2312-119X

АКТУАЛЬНІ ПРОБЛЕМИ АВТОМАТИЗАЦІЇ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ТОМ 20

Збірник наукових праць

Дніпро
ЛІРА
2016

УДК 519.689:519.254
ББК 32.973.202
А 43

Описано задачу автоматизованої обробки даних, інформаційну технологію сучасних обчислювальних процедур більш адекватного відображення експериментальних даних у системах моніторингу. Досліджено обчислювальні процедури сплайн-перетворень, які застосовані у системах ГІС та прийняття рішень, прикладні задачі інформаційних технологій.

Для науковців, аспірантів та фахівців у галузі інформатики.

А43 **Актуальні проблеми автоматизації та інформаційних технологій :** зб. наук. пр. / наук. ред. О. Г. Байбуз. Дніпро. 2016. Т. 20. 105 с.
ISSN 2312-119X

Описаны задачи автоматизированной обработки данных, информационная технология современных вычислительных процедур более адекватного отображения экспериментальных данных в системах мониторинга. Исследованы вычислительные процедуры сплайн-преобразований, примененных в системах ГИС и принятия решений, прикладные задачи информационных технологий.

Для научных сотрудников, аспирантов и специалистов в области информатики.

УДК 519.689:519.254
ББК 32.973.202

Свідоцтво про державну реєстрацію засобу масової інформації серія КВ № 5470 від 12.09.01
Рекомендовано вченою радою Дніпропетровського національного університету імені Олеся Гончара, протокол № 7 від 24.12.2015 р.

УДК 519.237.8:519.254:004.9

О. Г. Байбуз, М. Г. Сидорова, О. В. Лапец

Дніпропетровський національний університет імені Олеся Гончара

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ КЛАСТЕРНОГО АНАЛІЗУ РЕЗУЛЬТАТІВ ПСИХОЛОГІЧНОГО ТЕСТУ В УМОВАХ НЕВИЗНАЧЕНОСТІ

Запропоновано інформаційну технологію кластерного аналізу результатів міні-мульт тесту для визначення психологічних особливостей хворих на артеріальну гіпертензію. Технологія забезпечує підтримку прийняття рішень дослідником щодо вибору найякіснішого розв'язку в умовах невизначеності.

Ключові слова: *кластерний аналіз, інформаційна технологія, оцінка якості, міні-мульт тест.*

Предложена информационная технология кластерного анализа результатов мини-мульт теста для определения психологических особенностей больных артериальной гипертензией. Технология обеспечивает поддержку принятия решений исследователем по выбору качественного решения в условиях неопределенности.

Ключевые слова: *кластерный анализ, информационная технология, оценка качества, мини-мульт тест.*

The information technology of cluster analysis of mini-mult test results to determine the psychological characteristics of patients with arterial hypertension has proposed. Technology provides support for decision-making by researcher at the choice of quality solutions in the face of uncertainty.

Keywords: *cluster analysis, information technology, quality evaluation, mini-mult test.*

Вступ. Застосування кластерного аналізу є корисним у різноманітних предметних галузях, у тому числі і в медицині, і в психології, оскільки методи кластеризації дають змогу поділити сукупність об'єктів на однорідні за певним формальним критерієм подібності групи (кластери). Основною властивістю цих груп є те, що об'єкти, які належать одному кластеру, подібніші між собою, ніж об'єкти з різних кластерів. Таку класифікацію можна виконувати одночасно за досить великою кількістю ознак. Можна сформулювати такі цілі кластеризації:

- Розуміння даних шляхом виявлення кластерної структури.

Розбиття вибірки на групи схожих об'єктів дозволяє застосовувати до кожного кластера свій метод аналізу при подальшій обробці даних і прийнятті рішень.

- Стиснення даних. Якщо початкова вибірка надмірно велика, то можна скоротити її, залишивши по одному найбільш типовому представнику від кожного кластера.
- Виявлення новизни. Виділяються нетипові об'єкти, які не вдається приєднати ні до одного з кластерів.
- Формулювання та перевірка гіпотез на основі дослідження даних.

Аналіз літературних даних і постановка проблеми. Задачам кластерного аналізу приділено багато уваги [1–4]. Існують різні підходи і напрями досліджень, розроблено безліч методів та алгоритмів, багато дослідників розглядали дану тематику у своїх наукових роботах. Проте і досі існують питання, які не знайшли свого повного розв'язку (оцінка якості результатів, вибір методу та оптимальної кількості кластерів, визначення подібності об'єктів тощо). Якщо немає жодної експертної інформації щодо отримуваних кластерів та їх кількості, то досліднику необхідно приймати низку рішень в умовах невизначеності. Оскільки випадковий необґрунтований вибір методу та параметрів може призвести до того, що отримане розбиття буде зовсім відмінним від природної, притаманної досліджуваним даним, кластерної структури, актуальною задачею є розробка інформаційних технологій, що дозволять оцінювати якість отримуваних угруповань залежно від різних налаштувань параметрів та зменшити ризик вибору невідповідного розв'язку.

Постановка задачі. В роботі поставлено за мету провести кластерний аналіз даних, які є результатом дослідження когнітивних функцій хворих на артеріальну гіпертензію II і III стадії (з давністю перенесеного інсульту не менше 6 місяців). Було обстежено 46 пацієнтів, серед яких 24 чоловіки й 22 жінки, середній вік яких склав 46,7 року, тривалість артеріальної гіпертензії – 4,6 року. Для визначення психологічних особливостей хворих було використано міні-мульт тест, який являє собою скорочений варіант MMPI тесту та має 11 шкал (з них 3 – оцінні: відвертості, вірогідності, корекції, та 8 базисних: іпохондрії, депресії, істерії, психопатії, паранойяльності, психастенії, шизоїдності, гіпоманії), дійсні числа.

Необхідно розробити інформаційну технологію, що дозволить зрозуміти, яка кластерна структура притаманна досліджуваним даним,

проаналізувати отримані кластери, виявити схожість властивостей об'єктів аналізу, візуалізувати отримані результати для подальшої їх інтерпретації.

Основний матеріал. Для проведення кластерного аналізу результатів психологічного тесту при відсутності будь-якої допоміжної експертної інформації щодо отримуваних кластерів було розроблено інформаційну технологію, що складається з п'яти основних етапів.

1. *Попередня обробка даних*. Для зведення досліджуваних ознак до єдиного масштабу пропонується стандартизація, що обчислюється за формулою: $x'_{ij} = \dfrac{x_{ij} - \bar{x}_j}{\hat{\sigma}_j}, i = \overline{1,n}, j = \overline{1,p}$, де $\bar{x}_j = \dfrac{1}{n}\sum_{i=1}^{n} x_{ij}$, $\hat{\sigma}_j = \sqrt{\dfrac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}$. Після такого перетворення усі ознаки будуть мати однаковий вплив на результат кластеризації.

Важливим є проаналізувати інформативність досліджуваних ознак, з цією метою реалізовано метод апроксимації матриці відстаней та метод «гойдалки». Було виявлено, що усі 11 ознак є інформативними та мають бути розглянуті у подальшому аналізі.

2. *Визначення наявності кластерної структури*. Пропонується застосування критерію Дуда – Харта:

$$F_{DH} = w_2 / w_1,$$

де w_2 – сума квадратів внутрішньокластерних відстаней у випадку, коли дані розподілені на 2 кластери, w_1 – сума квадратів внутрішньокластерних відстаней у випадку одного кластеру.

Гіпотеза існування єдиного кластеру однорідних даних відхиляється, якщо значення критерію менше критичного значення, обчисленого за формулою:

$$F_{кр} = 1 - \frac{2}{\pi p} - u_{(1-\alpha)}\sqrt{2(1 - 8/(\pi^2 p))/(np)},$$

де $u_{(1-\alpha)}$ – квантиль стандартного нормального розподілу рівня $(1-\alpha)$.

Отримане значення критерію свідчить про те, що досліджувані дані не є однорідними та мають кластерну структуру.

3. *Застосування методів кластерного аналізу*. У загальному вигляді задача кластерного аналізу полягає у визначенні розбиття $G = \{g_1, g_2, \ldots, g_K\}$ на кластери множини $X = \{x_{ij}\}, i = \overline{1,N}, j = \overline{1,p}$, де

N – кількість об'єктів аналізу, p – кількість досліджуваних ознак, x_{ij} – значення j-ї ознаки і-го об'єкта, $g_i = \{x_l\}$, $i = \overline{1,K}$, $x_l = \{x_{lj}\}$, $l = \overline{1,N_i}$, $j = \overline{1,p}$, N_i – кількість об'єктів у і-му кластері, $\sum_{i=1}^{K} N_i = N$, $\bigcup_{i=1}^{K} g_i = X$, $g_i \bigcap g_j = \varnothing, i \neq j$.

Розглянемо декілька алгоритмів кластерного аналізу з тих, що склали ядро розробленого авторами програмного забезпечення.

Алгоритм агломеративної ієрархічної кластеризації.

1. Кожен об'єкт $x_i, i = \overline{1,N}$ вважаємо окремим кластером g_i, $N_i = 1$. Обираємо метрику та обчислюємо матрицю $D = \{d_{ij}\}, i, j = \overline{1,N}$, де d_{ij} – відстань між x_i та x_j.

2. У матриці відстаней D знаходимо мінімальний елемент d_{ij} і кластери g_i та g_j об'єднуємо $g_{i+j} = g_i \cup g_j$, $N_{i+j} = N_i + N_j$.

3. З матриці D вилучаємо відстані від g_i та g_j до інших кластерів та додаємо відстані, що відповідають новому кластеру g_{i+j}.

Для обчислення відстані між кластерами використовується загальна формула Ланса – Уільямса:

$$d(g_{i+j}, g_h) = \alpha_1 d(g_i, g_h) + \alpha_2 d(g_j, g_h) + \beta d(g_i, g_j) + \gamma \left| d(g_i, g_h) - d(g_j, g_h) \right|$$

Задаючи відповідні значення параметрів α_l, α_m, β, γ, отримаємо різні види агломеративних ієрархічних методів [4].

Повторюємо кроки 2–3, доки не отримаємо необхідну кількість кластерів або усі об'єкти не будуть об'єднані в один кластер для побудови дендрограми (рис. 2). На основі візуального аналізу дендрограми можна зробити висновок, що кластерна структура досліджуваних даних складається з двох груп. Це підтверджує і критерій різниці між рівнями дендрограми (рис. 1).

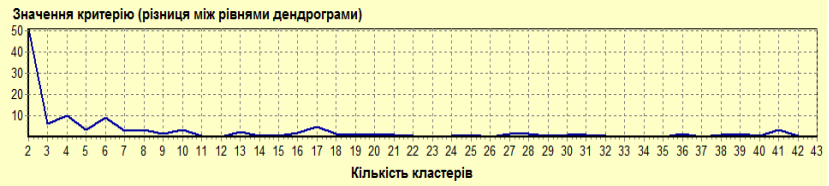


Рисунок 1 – Дослідження кількості кластерів на основі дендрограми

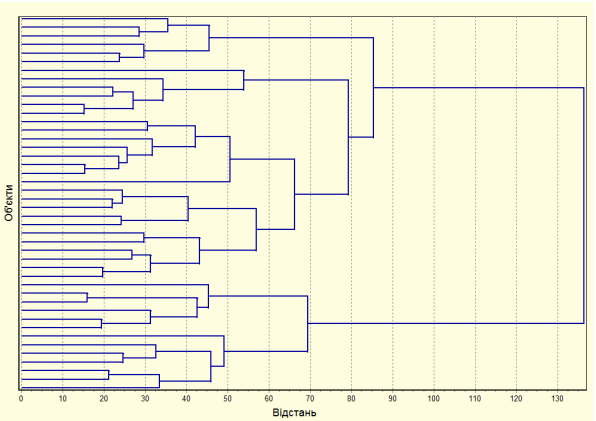


Рисунок 2 – Результат ієрархічної кластеризації у вигляді дендрограми

Методи нечіткої кластеризації [5–6] дозволяють одному і тому самому об'єкту належати одночасно кільком (або навіть усім) кластерам, але з різним ступенем приналежності. Таким чином, можна проаналізувати ступінь відокремленості кластерів один від одного та виявити їх перетини.

Алгоритм Уіндхема. Знаходитмо розв'язок оптимізаційної задачі

$$Q(P,V)=\sum_{l=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{N}\mu_{li}^{2}v_{lj}^{2}d(x_i,x_j)\to\min$$

у такому вигляді:

$$P^{*}=\arg\min_{P}\left\{\begin{matrix}Q(P,V):P=(M^{1},...,M^{K}),\\ M^{l}=\left[(\mu_{l1,...,}\mu_{lN}),(v_{l1,...,}v_{lN})\right]\end{matrix}\right\},$$

$0\le\mu_{li}\le 1$, $0\le v_{lj}\le 1$, $\sum_{l=1}^{K}\mu_{li}=1$, $\sum_{j=1}^{N}v_{li}=1$, $\sum_{i=1}^{N}\mu_{li}>0$, $\sum_{l=1}^{K}v_{lj}>0$,

$i,j=1,...,N$, $l=1,...,K$.

1. Випадковим чином задаємо матрицю розбиття $P_{K\times N}\in\left[\mu_{li}\right]$, $0\le\mu_{li}\le 1$, $\sum_{i=1}^{N}\mu_{li}>0$ та отримуємо початкове розбиття $P_{(0)}=(M_{(0)}^{1},...,M_{(0)}^{K})$ на K нечітких кластерів.

2. Покладаємо значення i = 1.

3. Покладаємо $P_{(i)}=P_{(0)}$ і обчислюємо матрицю прототипів $V_{(0)}$ таким чином:

$$v_{li} = \frac{1 \Big/ \sum_{i=1}^{n} \mu_{li}^2 d(x_i, x_j)}{\sum_{j=1}^{n} (1 \Big/ \sum_{i=1}^{n} \mu_{li}^2 d(x_i, x_j))}, \tag{1.1}$$

$i, j = 1,...,N$, $l = 1,...,K$.

4. Покладаємо $V_{(i+1)} = V_{(i)}$ і обчислюємо матрицю розбиття $P_{(i)}$ таким чином:

$$\mu_{li} = \frac{1 \Big/ \sum_{j=1}^{n} v_{lj}^2 d(x_i, x_j)}{\sum_{k=1}^{c} (1 \Big/ \sum_{j=1}^{n} v_{kj}^2 d(x_i, x_j))}, \tag{1.2}$$

$i, j = 1,...,N$, $l = 1,...,K$.

5. Обчислюємо матрицю прототипів $V_{(i)}$ на основі $P_{(i)}$ відповідно до співвідношення (1.1).

Якщо $Q(P_i, V_i) - Q(P_{i-1}, V_{i-1}) < \varepsilon$, то $P^* = P_{(i)}$, $V^* = V_{(i)}$, інакше i = i +1 і переходимо на крок 4.

Аналізуючи графік приналежності, який наведено на рис/ 3 (чим ближче до 1 значення функції приналежності об'єкта, тим з більшою вірогідністю можна віднести його відповідному кластеру), можна зробити висновок, що в цілому об'єкти чітко розподіляються на 2 кластера, і лише стосовно 2–3 об'єктів рішення про віднесення їх певному кластеру є менш однозначним.

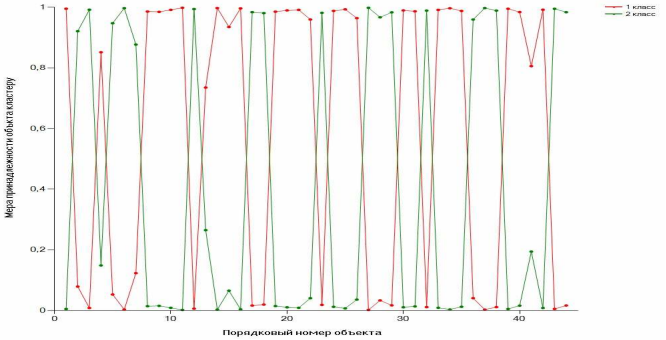


Рисунок 3 – Графік функції приналежності об'єктів дослідження нечітким кластерам

На рис. 4 результати кластеризації представлені у вигляді діаграми розсіювання (об'єкти дослідження зображуються точками у N-вимірному просторі та проектуються на площину). Візуально можна виділити 2 кластери з не досить чіткою межею.

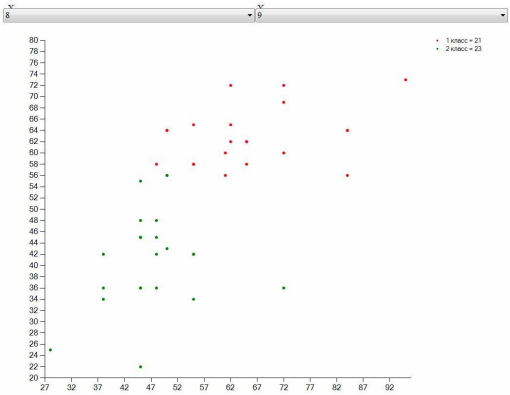


Рисунок 4 – Результат кластеризації у вигляді діаграми розсіювання

4. *Оцінювання якості, оптимальної кількості кластерів та вибір результуючого розв'язку.* На попередньому етапі ми отримали набір розв'язків $G = \{G_1, G_2, \ldots, G_n\}$, $G_i = \{g_1^{(i)}, g_2^{(i)}, \ldots, g_{K_i}^{(i)}\}$, серед яких необхідно виявити такий, що має найвищу якість. Оцінювання здійснюється за допомогою функціоналів якості, таких як сума внутрішньокластерних дисперсій за усіма ознаками, сума внутрішньокластерних відстаней, відношення внутрішньокластерних та міжкластерних відстаней; індексів Данна, Беждека – Данна, Калінського й Гарабача, а також на основі багатокритеріальних оцінок, отриманих із застосуванням колективних методів прийняття рішень [7–8].

Іноді декілька результатів кластеризації можуть представляти різні, але еквівалентні за якістю угруповання. У цьому випадку, замість того, щоб обирати один з розв'язків, пропонується використовувати ансамблевий підхід і отримати колективне результуюче рішення [9–10].

5. *Візуалізація і аналіз результатів.* Для аналізу та інтерпретації результатів розроблена система пропонує обчислення статистичних характеристик кожного кластера, а також широкий спектр засобів візуалізації. Було виявлено, що середні значення усіх ознак одного кластеру менші, ніж другого.

Висновки. У роботі запропоновано інформаційну технологію кластерного аналізу результатів міні-мульт тесту для визначення

психологічних особливостей хворих на артеріальну гіпертензію. Технологія забезпечує підтримку прийняття рішень дослідником щодо вибору найякіснішого розв'язку в умовах невизначеності. Досліджено кластерну структуру даних, виявлено два кластери, проаналізовано нечіткість приналежності об'єктів кожній з груп, а також статистичні характеристики усіх ознак за кожним кластером.

Бібліографічні посилання

1. Berkhin P. A survey of clustering data mining techniques // Grouping Multidimensional Data. 2006. P. 25–71.

2. Миркин Б. Г. Методы кластер-анализа для поддержки принятия решений: обзор. М. 2011. 88 с.

3. Бериков В.С., Лбов Г. С. Современные тенденции в кластерном анализе. Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению «Информационно-телекоммуникационные системы». 2008. 26 с.

4. Мандель И. Д. Кластерный анализ. М. 1988. 176 с.

5. Вятченин Д. А. Нечеткие методы автоматической классификации: монография. Мн. 2004. 219 с.

6. Oliveira J. V., Pedrycz W. Advances in fuzzy clustering and its applications. Chichester. 2007. 435 p.

7. Pascual D., Pla F., Sanchez J. S. Cluster validation using information stability measures // Pattern Recognition Letters. 2010. Vol. 31. P. 454–461.

8. Приставка О. П., Сидорова М.Г. Підтримка прийняття рішень в задачах кластерного аналізу // Актуальні проблеми автоматизації та інформаційних технологій : зб. наук. праць. Дніпропетровськ. 2011. Т. 15. С.117–125.

9. Сидорова М. Г. Застосування ансамблів алгоритмів для підвищення стійкості результатів кластеризації. // Актуальні проблеми автоматизації та інформаційних технологій : зб. наук. праць. Дніпропетровськ. 2013. Т. 17. С. 22 –29.

10. Pestunov I. A., Berikov V. B., Kulikova E. A. Rylov Ensemble of Clustering Algorithms for Large Datasets. // Optoelectronics, Instrumentation and Data Processing. 2011. Vol. 47. No. 3. P. 245–252.

Надійшла до редколегії 28.11.16.

УДК 519.688

А.О. Долгіх, О.І. Білобородько, О.Г. Байбуз

Дніпропетровський національний університет імені Олеся Гончара

ЗНАХОДЖЕННЯ ОПТИМАЛЬНИХ ЗНАЧЕНЬ ПАРАМЕТРІВ АДАПТИВНИХ МОДЕЛЕЙ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ З ВИКОРИСТАННЯМ ГЕНЕТИЧНОГО АЛГОРИТМУ

Описано алгоритм пошуку оптимальних значень коефіцієнтів згладжування адаптивних моделей прогнозування часових рядів, реалізований авторами. Запропонований алгоритм був застосований до вирішення задачі прогнозування економічних показників. Виконано аналіз результатів ефективності розробленого алгоритму.

Ключові слова*: прогнозування часових рядів, адаптивні моделі прогнозування, генетичний алгоритм з дійсним кодуванням, економічні часові ряди.*

Описан алгоритм поиска оптимальных значений коэффициентов сглаживания адаптивных моделей прогнозирования временных рядов, реализованный авторами. Предложенный алгоритм был применен к решению задач прогнозирования экономических показателей. Выполнен анализ результатов эффективности разработанного алгоритма.

Ключевые слова: *прогнозирование временных рядов, адаптивные модели прогнозирования, генетический алгоритм с вещественным кодированием, экономические временные ряды***.**

Optimal values of adaptive forecasting models smoothing factors search algorithm developed by authors has been described. The proposed algorithm has been applied to economical time series forecasting. Analysis of the developed algorithm efficiency has been done.

Keywords: *time series forecasting, adaptive forecasting models, real coded genetic algorithm, economical time series.*

Вступ. Особливої актуальності в різних галузях людської діяльності набуває задача прогнозування. В економіці – для прогнозування щоденних коливань цін на акції, курсів валют, щотижневих та щомісячних обсягів продаж, річних обсягів виробництва тощо. У природничих науках – для прогнозування кількості опадів, природних явищ, стану забруднення водних ресурсів, оцінки деяких біологічних

та біохімічних показників.

Для вирішення задачі прогнозування вдалим рішенням є застосування адаптивних моделей. Ці моделі дозволяють обчислювати прогнозні значення без перерахунку моделі у разі отримання нових даних, не накладають обмежень на розмір ряду та швидко підлаштовуються під динаміку змін рівнів ряду, ґрунтуються на принципі експоненціального згладжування, тим самим враховують «старіння» інформації [1].

Однак результати застосування цих методів сильно залежать від коефіцієнтів згладжування $\beta_1, \beta_2, \beta_3$. При зміні значень цих коефіцієнтів значення середньоквадратичної похибки може істотно зростати. Задача пошуку параметрів $\beta_1, \beta_2, \beta_3$ не має тривіального вирішення. Тому було вирішено реалізувати програмне забезпечення, яке б знаходило оптимальні параметри адаптивних моделей за допомогою генетичного алгоритму, та дослідити, яким чином параметри генетичного алгоритму впливають на якість побудованого прогнозу.

Аналіз літературних даних. Найбільш відомими моделями прогнозування часових рядів є регресійні моделі. У таких моделях ряд подають у вигляді суми або добутку трьох компонент:

$$f(t) = T(t) + S(t) + A(t) + \xi_t,$$

$$f(t) = T(t) * S(t) * A(t) * \xi_t \quad (1.1),$$

де $T(t)$ – лінія тренда, що показує глобальні зміни досліджуваного явища; $S(t)$ – сезонність, яка відображає коливання відносно тренда, обумовлені зовнішніми впливами; $A(t)$ – циклічність (автоколивання) – більш-менш регулярні коливання відносно тренда, обумовлені внутрішньою природою досліджуваного явища.

Задача прогнозування часових рядів у випадку регресійних моделей зводиться до задачі виділення цих трьох компонент. Трендову компоненту представляють у формі лінійної чи нелінійної регресії, параметри якої шукають методом найменших квадратів. Для подання сезонної та циклічної складової використовують один з двох підходів: індекси сезонності або гармонічний аналіз [2]. Для побудови регресійних моделей необхідно мати велику базу даних спостережень. Корисно також мати незалежний набір прикладів, на яких можна перевірити якість роботи моделі. Окрім цього, окремою проблемою при використанні цих моделей є обрання вигляду лінії регресії.

Сьогодні великої популярності набувають методи, які

представляють собою подальший розвиток методу регресійного аналізу, зокрема метод МГУА – метод групового урахування аргументів. Метод реалізує задачу синтезу оптимальних моделей високої складності, адекватної складності досліджуваного об'єкта (тут під моделями розуміється система регресійних рівнянь). Алгоритми МГУА, побудовані за схемою масової селекції, здійснюють перебирання можливих функціональних описів об'єкта. До явних переваг МГУА належать автоматичне формування структури мережі, простота і швидкодія настроювання параметрів, а також можливість «згорнути» налаштовану мережу безпосередньо в явний математичний вираз. Головною проблемою при використанні цього методу є те, що для якісного результату цей метод, як і всі регресійні моделі, вимагає досить великої репрезентативної вибірки та складних математичних обчислень [3].

Другим великим класом є моделі прогнозування, які базуються на принципі згладжування, тобто враховують «старіння» інформації. Згладжування часових рядів проводять з метою виділення присутніх у ряді тенденцій та видалення аномальних значень. Одним з найпростіших і найпоширеніших підходів є експоненційне згладжування, в основі якого лежить розрахунок еспоненційних середніх. На основі цього підходу було розроблено багато моделей прогнозування, зокрема адаптивні моделі, головними перевагами яких у порівнянні з регресійними методами є те, що вони не накладають обмежень на розміри ряду та не потребують виконання складних математичних операцій. Ці моделі є основним об’єктом поточного дослідження.

Варто зауважити, що сьогодні існує широкий клас методів, які використовують сучасні методи машинного навчання: нейронні мережі, дерева рішень, марківські моделі для вирішення задачі прогнозування часових рядів.

Основний матеріал. В адаптивних методах часовий ряд подають у вигляді функції

$$u_t = f\left(a_{1t}, a_{2t}, ..., a_{pt}, t\right) + e_t \,, \tag{1.2}$$

під час побудови якої відслідковують величину відхилень прогнозних значень від значень вихідного ряду. Адаптивні методи прогнозування поділяються на лінійні та сезонні. У моделях лінійного зростання прогноз обчислюють за формулою:

$$u_{t+\tau} = \hat{a}_{1,t} + \hat{a}_{2,t}\tau \,, \tag{1.3}$$

де τ – кількість кроків прогнозу; $\hat{a}_{1,t}$, $\hat{a}_{2,t}$ – коефіцієнти адаптивної

моделі в момент часу t, $\hat{a}_{1,t}$ – оцінка того, чого досягли на поточному кроці; $\hat{a}_{2,t}$ – приріст на поточному кроці.

До адаптивних моделей лінійного зростання відносять модель Хольта, модель Тейла – Вейджа, модель Брауна та модель Бокса – Дженкінса. Серед сезонних моделей виділяють лінійну адаптивну, лінійну мультиплікативну, експоненційну адитивну та експоненційну мультиплікативну. Вказані моделі відрізняються засобами знаходження параметрів $\hat{a}_{1,t}$, $\hat{a}_{2,t}$. Формули оцінок параметрів за адаптивними лінійними моделями наведено у табл. 1. $\beta_1, \beta_2, \beta_3$ – коефіцієнти згладжування, які приймають значення від 0 до 1.

Таблиця 1 – Адаптивні моделі лінійного зростання

Модель	**Формула підрахунку оцінок $\hat{a}_{1,t}$ та $\hat{a}_{2,t}$**
Модель Хольта	$\hat{a}_{1,t} = \beta_1 u_t + (1-\beta_1)(\hat{a}_{1,t-1} + \hat{a}_{2,t-1}),$ $\hat{a}_{2,t} = \beta_2 (\hat{a}_{1,t} - \hat{a}_{1,t-1}) + (1-\beta_2)\hat{a}_{2,t-1},$
Модель Тейла – Вейджа	$\begin{cases} \hat{a}_{1,t} = \beta_1 u_t + (1-\beta_1)\hat{u}_t = \hat{u}_t + \beta_1 e_t, \\ \hat{a}_{2,t} = \hat{a}_{2,t-1} + \beta_1\beta_2 e_t, \end{cases}$ $e_t = u_t - \hat{u}_t$
Модель Брауна	$\begin{cases} \hat{a}_{1,t} = \hat{u}_t + (1-\beta^2) e_t, \\ \hat{a}_{2,t} = \hat{a}_{2,t-1} + (1-\beta^2) e_t. \end{cases}$ $e_t = u_t - \hat{u}_t$
Модель Бокса – Дженкінса	$\begin{cases} \hat{a}_{1,t} = \hat{u}_t + \beta_1 e_t + \beta_3 (e_t - e_{t-1}), \\ \hat{a}_{2,t} = \hat{a}_{2,t-1} + \beta_1\beta_2 e_t. \end{cases}$ $e_t = u_t - \hat{u}_t$

У табл. 2 подано схему обчислень для адаптивних сезонних моделей. У наведених моделях сезонність – це масиви довжиною l $g_{1,\dots,}g_l$, $f_{1,\dots,}f_l$, де l – період сезонності. Елементи цих масивів обчислюють на кожному кроці, як і коефіцієнти моделей.

Таблиця 2 – Адаптивні моделі сезонного зростання

Модель	**Формула підрахунку оцінок $\hat{a}_{1,t}$ та $\hat{a}_{2,t}$**
Лінійна адитивна	$\hat{u}_{t+\tau} = \hat{a}_{1,t} + \hat{a}_{2,t} \cdot \tau + \hat{g}_{t+\tau-l}$ $\hat{a}_{1,t} = \beta_1 \left(u_t - \hat{g}_{t-l}\right) + \left(1-\beta_1\right)\left(\hat{a}_{1,t-1} - \hat{a}_{2,t-1}\right)$ $\hat{a}_{2,t} = \beta_2 \left(\hat{a}_{1,t} - \hat{a}_{1,t-1}\right) + \left(1-\beta_2\right)\hat{a}_{2,t-1},$ $\hat{g}_t = \beta_3 \left(u_t - \hat{a}_{1,t}\right) + \left(1-\beta_3\right)\hat{g}_{t-l}.$
Лінійна мультиплікативна (модель Уінтерса)	$\hat{u}_{t+\tau} = \left(\hat{a}_{1,t} + \hat{a}_{2,t} \cdot \tau\right) \cdot \hat{f}_{t+\tau-l}$ $\hat{a}_{1,t} = \beta_1 \frac{u_t}{\hat{f}_{t-l}} + \left(1-\beta_1\right)\left(\hat{a}_{1,t-1} + \hat{a}_{2,t-1}\right),$ $\hat{a}_{2,t} = \beta_2 \left(\hat{a}_{1,t} - \hat{a}_{1,t-1}\right) + \left(1-\beta_2\right)\hat{a}_{2,t-1},$ $\hat{f}_t = \beta_3 \frac{u_t}{\hat{a}_{1,t}} + \left(1-\beta_3\right)\hat{f}_{t-l}.$
Експоненційна адитивна	$\hat{u}_{t+\tau} = \hat{a}_{1,t} \cdot \hat{r}_{1,t}^{\tau} + \hat{g}_{t+\tau-l}$ $\hat{a}_{1,t} = \beta_1 \left(u_t - \hat{g}_{t-l}\right) + \left(1-\beta_1\right)\left(\hat{a}_{1,t-1} \cdot \hat{r}_{t-1}\right),$ $\hat{r}_t = \beta_2 \frac{\hat{a}_{1,t}}{\hat{a}_{1,t-1}} + \left(1-\beta_2\right)\hat{r}_{t-1},$ $\hat{g}_t = \beta_3 \left(u_t - \hat{a}_{1,t}\right) + \left(1-\beta_3\right)\hat{g}_{t-l}.$
Експоненційна мультиплікативна	$\hat{u}_{t+\tau} = \left(\hat{a}_{1,t} \cdot \hat{r}_{1,t}^{\tau}\right) \cdot \hat{f}_{t+\tau-l}$ $\hat{a}_{1,t} = \beta_1 \frac{u_t}{\hat{f}_{t-l}} + \left(1-\beta_1\right)\left(\hat{a}_{1,t-1} \cdot \hat{r}_{t-1}\right),$ $\hat{r}_t = \beta_2 \frac{\hat{a}_{1,t}}{\hat{a}_{1,t-1}} + \left(1-\beta_2\right)\hat{r}_{t-1},$ $\hat{f}_t = \beta_3 \frac{u_t}{\hat{a}_{1,t}} + \left(1-\beta_3\right)\hat{f}_{t-l}.$

Характерною ознакою усіх адаптивних моделей є те, що на кожному кроці обчислюються нові значення a_{1t}, a_{2t}, які залежать від коефіцієнтів згладжування $0 < \beta_1, \beta_2, \beta_3 < 1$. Для того щоб отримати якісні результати прогнозування, потрібно підібрати такі параметри $\beta_1, \beta_2, \beta_3$, які найбільш підходять для заданого часового ряду. Для вирішення цієї задачі було реалізовано програмне забезпечення, яке б дозволяло знаходити оптимальні значення $\beta_1, \beta_2, \beta_3$ за допомогою генетичного алгоритму.

Потенційне рішення оптимізаційної задачі (параметри $\beta_1, \beta_2, \beta_3$) представлено у вигляді хромосом. У класичних генетичних алгоритмах хромосоми подаються у вигляді бінарного вектора. Однак бінарне подання хромосом тягне за собою певні труднощі при виконанні пошуку в безперервних просторах, які пов'язані з великою розмірністю простору пошуку [4]. Тому для вирішення поставленої задачі було використано генетичний алгоритм з кодуванням на базі дійсних чисел (*RGA*).

У ході виконання дослідження було реалізовано такі оператори схрещування неперервних генетичних алгоритмів: арифметичний, геометричний, плоский, лінійний, дискретний, евристичний, *BLX* – *α* та *SBX* кросовери та два види мутації: проста мутація та мутація Міхалевича. Також передбачено можливість обрати одну із стратегій відбору батьків для схрещування: панміксія, аутбридинг, інбридинг, турнірна селекція та метод рулетки. Розроблений програмний продукт дозволяє комбінувати різні оператори відбору, схрещування та мутації та відслідковувати, як при цьому змінюється середньоквадратична похибка прогнозу *MSE*. Користувач може змінювати розмір популяції, ймовірність мутації, кількість епох алгоритму, а також параметри, специфічні для деяких операторів схрещування дійсних векторів.

Прогнозування часових рядів з використанням адаптивних методів та генетичного алгоритму.

Крок 1. На першому кроці генерують початкову популяцію хромосом. Обсяг популяції N задається користувачем. Кожна хромосома представляє собою набір значень $\beta_1, \beta_2, \beta_3$. Для деяких лінійних моделей прогнозування, наприклад, моделі Тейла – Вейджа і Хольта, значення останнього параметра β_3 не використовується. Значення тих параметрів, які не використовується тією чи іншою моделлю, тотожно дорівнюють 0.

Крок 2. Розраховують значення середньоквадратичної похибки прогнозу за формулою:

$$MSE = \frac{\sum_{j=1}^{n}\left(ForecastedValue_j - ActualValue_j\right)^2}{n} \quad (1.4)$$

для кожної хромосоми в популяції. Для цього, використовуючи параметри $\beta_1, \beta_2, \beta_3$, за обраною користувачем адаптивною моделлю будують прогнозний ряд. Після цього розраховують значення середньоквадратичної похибки прогнозу *MSE*. Чим менше значення *MSE*, тим краще побудована модель підлаштовується під значення заданого ряду.

Крок 3. Обирають дві хромосоми з популяції за тією стратегією, яка задана користувачем, і проводять схрещування хромосом. У результаті цієї операції отримують дві нові особини, які успадковують ознаки батьків. Обох «батьків» та обох «нащадків» додають до їх «проміжної» популяції. Відбір батьків та схрещування проводять M разів. M задається користувачем.

Крок 4. Проводять мутацію. Для кожної хромосоми з популяції генерують деяке випадкове число. Якщо воно менше рівня мутації, то проводять мутацію хромосоми за однією з можливих стратегій: проста мутація або мутація Міхалевича. Вид мутації також обирається користувачем.

Крок 5. На цьому кроці проводять селекцію або відбір хромосом до наступного покоління. Для отриманої «проміжної» популяції розраховують значення *MSE*. Деяка кількість хромосом з найменшим значенням середньоквадратичної похибки прогнозу q потрапляє до наступної популяції. Задля того щоб попередити завчасне виродження популяції, випадковим чином генерують ще $N - q$ хромосом та додають їх до нової популяції.

Кроки 2–5 проводять ітеративно, поки не виконається одна з умов зупинки генетичного алгоритму: дотягнеться введена користувачем кількість поколінь або алгоритм зійдеться до єдиного рішення.

Після цього отриманий набір значень коефіцієнтів $\beta_1, \beta_2, \beta_3$ використовують для прогнозування наступних значень ряду.

Результати прогнозування з використанням адаптивних методів, налагоджених за допомогою генетичних алгоритмів.

Дані, які підлягають аналізу та прогнозуванню, представляють собою *ask price* на акції компанії AAON inc. в період з 4 січня 2012 року по 7 листопада 2014 року. *Ask price* – це ціна, за якою власник акції готовий її продати. Довжина ряду, $N = 282$. На рис. 1 зображено вхідні дані на графіку.

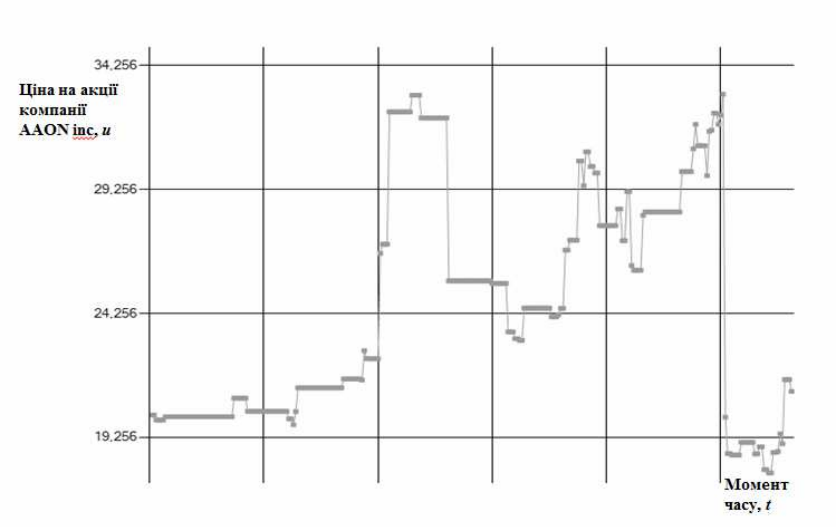


Рисунок 1 – Вхідні дані

У результаті перевірки вхідних даних на випадковість за допомогою критерію Спірмена та серіальної кореляції було встановлено, що введена послідовність не є випадковою, має тенденцію до зростання, а також у ній наявна сезонна складова.

Результати прогнозування досліджуваного ряду за допомогою адаптивних моделей та реалізованого алгоритму пошуку оптимальних значень коефіцієнтів згладжування представлено у табл. 3.

Для налагоджування моделей застосовувався генетичний алгоритм з дійсним кодуванням з такими параметрами: N = 50 (розмір популяції), відсоток кращих хромосом, які потрібно залишити популяції, q = 80 %, кількість епох (повторень) генетичного алгоритму = 50, рівень мутації = 0,1. Використовувалося значення K = 3. Для сезонних моделей значення довжини сезонності l = 12. Реальне значення ряду U_{n+1} = 21,11.

Таблиця 3 – Результати роботи реалізованого алгоритму пошуку оптимальних значень коефіцієнтів згладжування

Модель	β_1	β_2	β_3	***MSE***	**Побудоване прогнозне значення**
Модель Брауна	0,5036	N/A	N/A	1,4139	21,6228
Модель Хольта	0,9841	0,0039	N/A	1,2103	21,0745
Модель Тейла – Вейджа	0,9987	0,0126	N/A	1,2143	21,0616
Модель Бокса – Дженкінса	0,9666	0,0031	0,0479	1,2066	21,048
Лінійна адитивна	0,5836	0,5358	0,3059	1,221	21,1118
Лінійна мультиплікативна	0,0176	0,0065	0,9794	1,2238	21,125
Експоненційна адитивна	0,0033	0,0109	0,9688	1,2216	21,1104
Експоненційна мультиплікативна	0,0061	0,0151	0,9803	1,2206	21,1106

Задача знаходження оптимальних значень параметрів $\beta_1, \beta_2, \beta_3$ може бути розглянута як задача оптимізації в багатовимірному просторі. В даному випадку потрібно знайти такі значення $\beta_1, \beta_2, \beta_3$, при яких *MSE* приймає мінімальне значення. Тому програма надає можливість побудувати графік поверхні, для якої шукають екстремум (рис. 2). У якості координат *X* та *Y* використовують значення β_1, β_2 або β_3. У якості координати *Z* виступає значення *1/ MSE*. Ділянка, де *1/ MSE* досягає максимального значення, позначена на графіку червоним кольором.

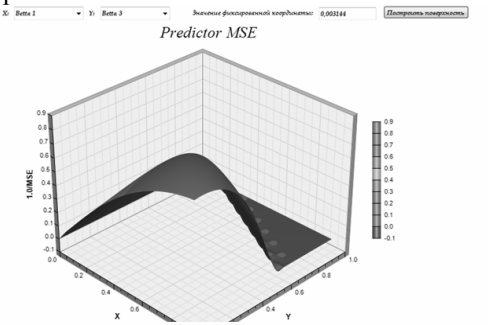


Рисунок 2 – Поверхня для моделі Бокса – Дженкінса

У табл. 4 наведено залежність середньоквадратичної похибки прогнозу *MSE* від кількості точок прогнозу *M*. Значення коефіцієнтів згладжування $\beta_1, \beta_2, \beta_3$ для кожної з моделей тотожно дорівнюють значенням у табл. 3.

Таблиця 4 – Залежність MSE від кількості точок прогнозу

Модель	***MSE* для *M*=1**	***MSE* для *M*=2**	***MSE* для *M*=3**	***MSE* для *M*=4**	***MSE* для *M*=5**	***MSE* для *M*=6**	***MSE* для *M*=7**
Модель Хольта	1,2	1,2	1,2	1,2	1,297	1,268	1,339
Модель Брауна	1,4	1,42	1,44	1,445	1,517	1,41	1,46
Модель Тейла – Вейджа	1,2	1,2	1,21	1,21	1,308	1,278	1,36
Модель Бокса – Дженкінса	1,2	1,2	1,2	1,2	1,3	1,263	1,336
Лінійна адитивна модель	1,2	1,2	1,2	1,24	1,287	1,212	1,28
Лінійна мультиплікативна модель	1,21	1,24	1,25	1,274	1,304	1,306	1,306
Експоненційна адитивна модель	1,2	1,242	1,259	1,285	1,316	1,323	1,32
Експоненційна мультиплікативна модель	1,2	1,241	1,258	1,283	1,315	1,322	1,323

Нижче подано графік залежності значення MSE прогнозу від кількості точок прогнозу для двох лінійних моделей : Хольта та Бокса – Дженкінса та двох сезонних моделей: лінійної адитивної та експоненційної мультиплікативної.

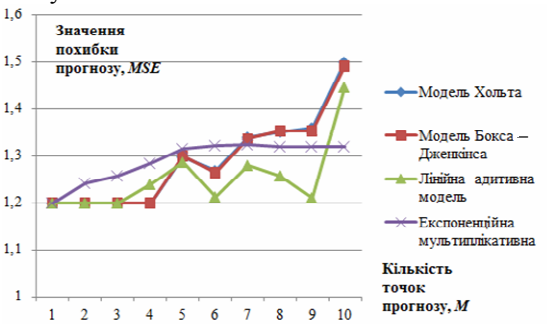


Рисунок 3 – Графік залежності MSE від кількості точок прогнозу

Як видно з графіка та табл. 4, значення середньоквадратичної похибки прогнозу починає істотно збільшуватися при кількості точок прогнозу більше трьох – чотирьох.

Висновки. Адаптивні моделі прогнозування мають ряд переваг у порівнянні з багатьма іншими відомими моделями, серед яких головним є те, що вони не вимагають суттєвих математичних обчислень, є простими у реалізації, не накладають обмежень на довжину ряду та дозволяють швидко побудувати прогноз.

На досліджуваних часових рядах розроблене програмне забезпечення показало найкращі результати прогнозування при використанні моделей Хольта, Тейла – Вейджа, Бокса – Дженкінса та адитивної лінійної моделі з такими параметрами генетичного алгоритму: використання арифметичного кросовера або *SBX – n* кросовера за параметром $n = 5$, аутбридингу, розмірності популяції 50 особин, рівня мутації = 0,1 та q = 80 %. В результаті проведення експериментів було встановлено, що потрібна кількість епох алгоритму залежить від розмірності ряду, якщо ряд є коротким, то задля знаходження оптимального рішення потрібно більше повторень, ніж для середніх та довгих. Окрім цього, було помічено, що алгоритм пошуку оптимальних параметрів швидше сходиться для моделей, які мають меншу кількість параметрів. Наприклад, для того щоб знайти оптимальний параметр для моделі Брауна, потрібно 10–15 епох, у той же час для того щоб налаштувати модель Бокса – Дженкінса, яка використовує три параметри, потрібно в середньому 30–40 повторень.

Також було встановлено, що з використанням реалізованого підходу якісний прогноз можна побудувати на три – чотири кроки вперед. При прогнозуванні на більшу кількість кроків вперед, середньоквадратична похибка прогнозу починає різко збільшуватися і побудований прогноз стає неадекватним.

Досліджувані моделі також мають і деякі недоліки, які, насамперед, викликані особливістю усіх адаптивних моделей – вони будують прогнозне значення із запізненням на один крок. У випадках, коли має місце різкий спад чи різкий зріст рівнів ряду, або коли знак тенденції змінюється зі зросту на спад чи зі спаду на зріст, такі моделі не можуть спрогнозувати майбутні значення правильно. Тому метою подальших досліджень є пошук методів, які були б більш гнучкими, могли аналізувати часовий ряд на присутні в ньому тенденції та шаблони і, таким чином, будували більш точний прогноз або надавали вірогідність декількох можливих майбутніх значень.

Бібліографічні посилання

1. Лукашин Ю. П. Адаптивные методы краткосрочного прогнозирования временных рядов: учебное пособие. Москва. 2003. 416 с.

2. Білобородько О. І., Ємельяненко Т. Г. Аналіз динамічних рядів: навчальний посібник. Дніпропетровськ. 2014. 80 с.

3. Метод групового урахування аргументів (МГУА) / URL: http://www.mgua.irtc.org.ua/ru/index.php?page=gmdh (дата звернення 5.11.2016).

4. Herrera F., Lozano M., Verdegay J. L. Tackling real-coded Genetic algorithms: operators and tools for the behaviour analysis // Artificial Intelligence Review. Vol. 12. No. 4. 1998. P. 265–319.

Надійшла до редколегії 05.11.16.

УДК 004.4:004.932(045)

В. М. Курочкін

Національний авіаційний університет

ДОСЛІДЖЕННЯ ІНФОРМАТИВНОСТІ КОЛЬОРОВИХ СКЛАДОВИХ ЗОБРАЖЕННЯ В ІНФОРМАЦІЙНІЙ СИСТЕМІ RANGER

Розповсюдженою кольоровою схемою, що використовується в цифрових зображеннях, є RGB, що має три складові. Проте для різних потреб інформативність ознак відрізняється. З метою оптимізації обчислень доречним є використання найбільш інформативних складових та різних кольорових моделей. Таким чином, було встановлено, що для аналізу зображень рослинності найбільш інформативною є синя складова кольорової системи RGB та H складова моделі HSI.

Ключові слова*: Digital image, обробка зображень, інформаційна технологія, кольорова модель, RGB, YIQ, HSI.*

Распространенной цветной схемой, что используется в цифровых изображениях, является RGB, состоящая из трёх компонентов. Но для разных прикладных задач их информативность может различаться. В целях оптимизации вычислений имеет смысл использовать наиболее информативные составляющие различных цветовых моделей. Таким образом, было установлено, что для анализа изображений растительности наиболее информативной будет синяя составляющая модели RGB и H составляющая модели HSI.

Ключевые слова*: цифровое изображение, обработка изображений, информационная технология, цветная модель, RGB, YIQ, HSI.*

A common color model used in digital images are RGB that consists of three components. However, for different uses the informational content may differ as well. To optimize the calculations and achieve better result, the most informative components of different color models may be used. Thus, in the article above it was found that the most informative when dealing with vegetation images is to use blue component of RGB color system and hue component of HIS model.

Keywords*: Digital image, digital imaging, informational technology, color model, RGB, YIQ, HSI.*

Постановка проблеми. З розвитком цифрової фотозйомки та доступної авіації усе актуальнішим стає обробка і аналіз цифрового зображення та відео. З використанням зображення високого розділення обсяги даних, що потребують обробки, спричиняють проблеми зі швидкодією аналізу, що посилюється наявністю трьох кольорових складових найпоширенішого цифрового формату RGB [1], якщо використовувати усі три складові. На практиці частіше зустрічається обробка монохромного зображення, коли використовується лише червона складова зображення, або усереднене значення інтенсивності кольорових складових. Проте при такому підході не беруться до уваги особливості інформативності кольорових складових для конкретних задач, що може спричинити погіршення результатів аналізу, що особливо важливо під час кластерного аналізу посівних площ [2] та аналізу розподілів інтенсивності кольорових складових для оцінки врожайності [3].

Таким чином, постає задача вивчення інформативності кольорових систем та складових для аналізу даних аерофотозйомки посівних територій методами кластерного аналізу для оптимізації обчислювальної складності обробки.

Аналіз публікацій. Колірний простів RGB складається з трьох складових: червоний, зелений і синій, що є рівноправними, проте у відтінках сірого світлі тони мають вищу інтенсивність, а кольори червоний, зелений і синій є дуже корельованими [4], що ускладнює роботу алгоритмів обробки зображень.

Кольорова модель має відповідати цілям обробки. Вибір найкращого уявлення кольору включає в себе аналіз того, як генерується сигнал та яку інформацію з цього сигналу необхідно вилучити. Загалом кольорові моделі можуть використовуватися для визначення кольору, розрізнення кольорів, виділення схожості між кольорами та визначення категорій кольорів тощо. Загальна класифікація кольорових систем включає такі: пристрій-орієнтовані моделі, користувач-орієнтовані та пристрій-незалежні кольорові моделі [8]. Буде логічним припустити, що найбільш значущими для обробки зображень є пристрій-незалежні кольорові моделі.

Одним з рішень проблеми інформативності кольорових складових RGB є простори кольорів.

YUC, YIQ, YCrCb, HSV, HSL тощо, де основне інформаційне навантаження несе люмінесцентна складова (яскравість зображення), а дві кольорові складові доповнюють її. Для отримання значень YIQ використовуються такі формули [1]:

$$\begin{cases} Y = 0.299R + 0.587G + 0.114B, \\ I = 0..529R - 0.275G - 0.3216B, \\ Q = 0.212R - 0.538G + 0.311B, \end{cases}$$

і, відповідно,

$$\begin{cases} R = Y + 0.956I + 0.621Q, \\ G = Y - 0.272I - 0.647Q, \\ B = Y - 1.1071I + 1.704Q, \end{cases}$$

У той час, як перетворення в кольорову модель HSI є більш обчислювально складним:

$$H = \begin{cases} \Theta, B \le G, \\ 360 - \Theta, B > G, \end{cases}$$

$$\Theta = \arccos\left\{ \frac{1/2\left[(R-G)+(R-B)\right]}{\left[(R-G)^2 + (R-B)(G-B)\right]^{1/2}} \right\},$$

$$S = 1 - \frac{3}{(R+G+B)}\left[\min(R,G,B)\right],$$

$$I = \frac{1}{3}(R+G+B),$$

перетворення з HSI в RGB залежить від значення H і представлено в [5].

Таким чином, використовуючи моделі такого типу, є можливість проводити окремо аналіз яскравості, тону та насиченості, що для аналізу є більш інформативним, ніж частки кольорів в суміші пікселю.

Проте пошуки оптимального уявлення кольорового зображення усе ще є актуальними, так, на базі HSI розробляються нові кольорові схеми [6], [7], для використання в обробці та аналізі цифрових зображень з більшою ефективністю та швидкодією.

У роботі [9] проведено порівняльний аналіз використання різних кольорових моделей у поєднанні з деякими кольоровими дескрипторами при проведенні процесу класифікації над зображенням. Результати показали, що найкращий результат отримано при використанні CCV [10] та моментів первинного статистичного аналізу.

Постановка задачі. Мета даної статті – дослідження впливу вибору кольорової моделі та конкретних складових моделей на результати аналізу цифрового зображення.

Основний матеріал. Для дослідження інформативності складових

кольорових схем при аналізі цифрового зображення було побудовано програмне забезпечення Ranger мовою програмування Java 8.

Програмне забезпечення складається з двох екранів – головного, призначеного для завантаження зображення, та виділення певної ділянки для більш ретельного вивчення у другому екрані дослідження (рис. 1).

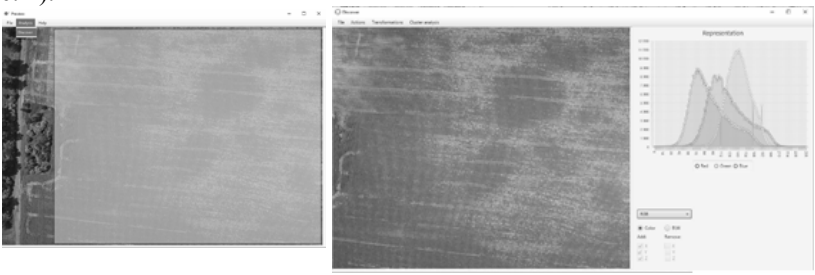


Рисунок 1 – Головне вікно та вікно дослідження програмного забезпечення Ranger

Основні можливості Ranger зводяться до вибору певних кольорових моделей (RGB, YIQ, HSI) та виокремлення кольорових складових, відображення гістограми розподілу інтенсивностей та відображення монохромного зображення на основі вибраних моделей і складових. Для тестування також реалізовано ряд перетворень та проста кластеризація [2] для порівняння результатів згідно з поставленими цілями.

Отже, нехай маємо зображення деякої посівної ділянки (рис. 2).

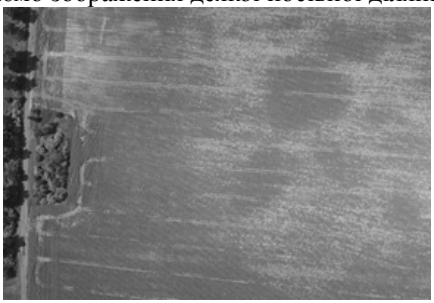


Рисунок 2 – Ділянка поля, що підлягає аналізу

Необхідно дослідити ефективність аналізу складових моделей RGB, YUV, HIS для відокремлення насичених рослинністю зон від пустих для визначення рівня прогнозованої врожайності території та виділення зон, що потребують додаткової уваги фермерів, наприклад, пересівання.

Розглянемо зображення (рис. 3) та гістограми (рис. 4) окремих кольорових складових, усереднених значень різних кольорових моделей.

Наочно можна побачити, що зелена складова дає найбільш тьмяне зображення і гістограма найбільш наближена до нормального розподілу, в той час як червона і синя складові дають більш контрастне зображення і їх гістограми мають більш виражені ознаки суміші двох нормальних розподілів та більше значення квадратичного відхилення. Усереднене зображення є очікувано контрастнішим за зелену складову, проте тьмянішим за червону і синю, маючи теж виражену суміш двох розподілів, проте згладжену через зелену компоненту. Різний зсув відносно середнього значення кольору також призводить до згладження розподілу. Y компонента незначно відрізняється від усереднення складових RGB, а I складова моделі HSI власне і дорівнює цьому усередненню. Складові H та S моделі HSI дають додаткову інформацію, проте за гістограмою важко зробити припущення про їх вплив. Схожі гістограми можна отримати при використанні I та Q складових YIQ, що пропущені в рис. 4 через їх неінформативність.

При проведенні кластеризації зображень (рис. 5) також можна помітити таке:

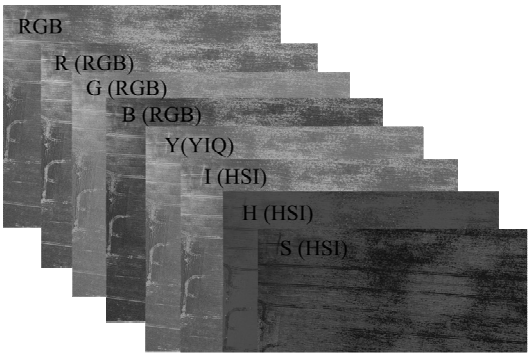


Рисунок 3 – Монохромні зображення, створені за допомогою усереднення та виділення окремих складових кольорових моделей

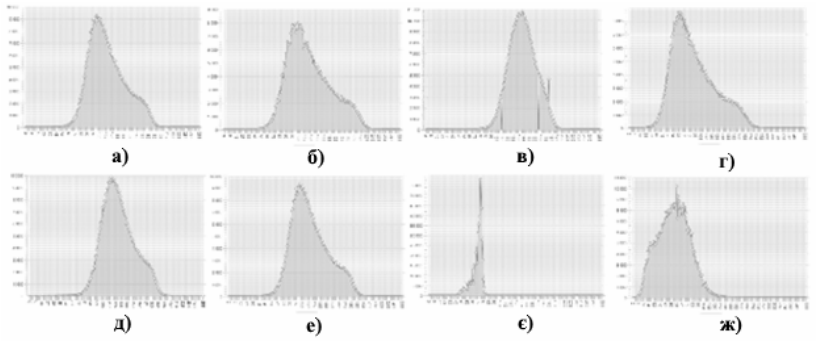


Рисунок 4 – Гістограми розподілу значень кольорових складових кольорових моделей RGB (а – усереднення, б, в, г – складові), YIQ (д – Y складова), HSI (е – I, є – S, ж – H)

1) При усередненні (рис. 5 а) отримуємо найбільш загальні результати з мінімальними деталями.

2) При проведенні кластеризації усіх трьох компонент моделі RGB (рис. 5 б), крім збільшеного часу обробки, ми отримуємо перенасичену інформацію, велику кількість класів, надмірну деталізацію.

3) Як і очікувалося, кластеризація червоної (рис. 5 в) і синьої (рис. 5 д) складових дає найбільш оптимальну деталізацію.

4) Зелена складова (рис. 5. г) дає найменш значущі результати.

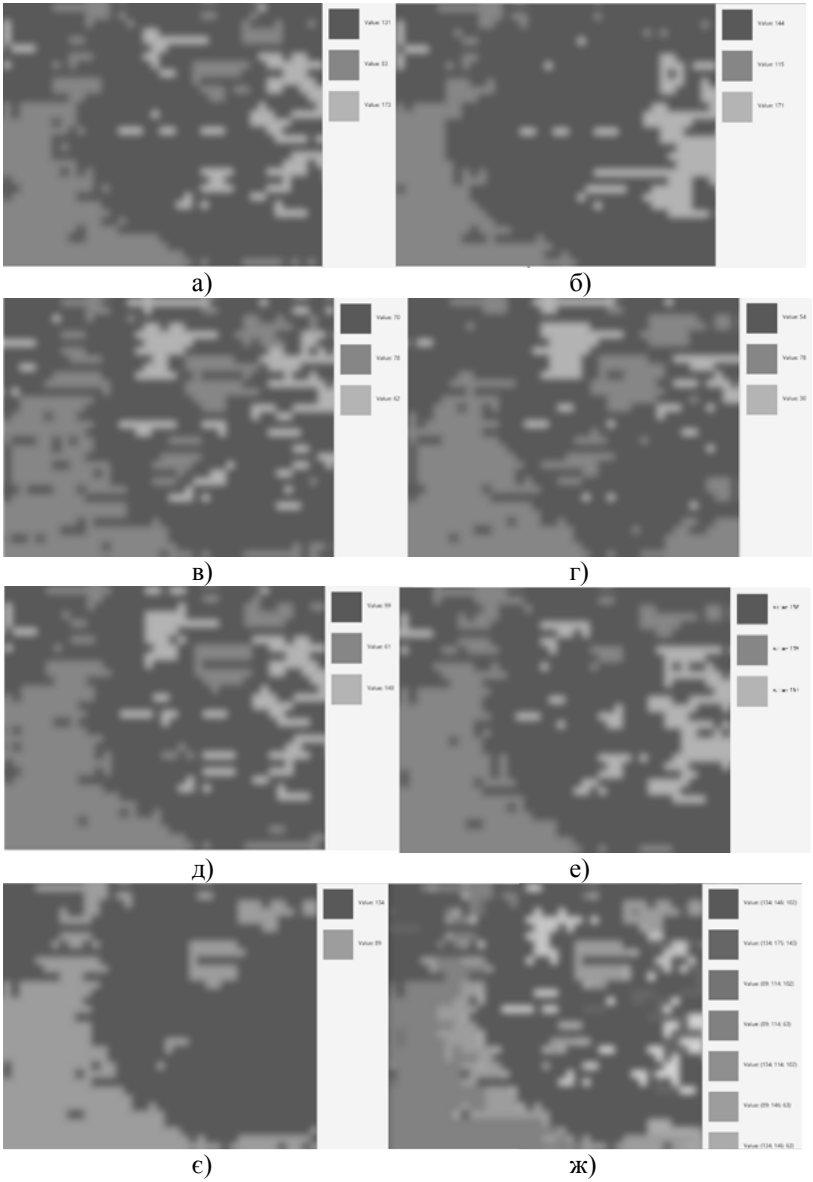
5) Найповніший результат отримується при поєднанні результатів аналізу складової Y моделі YIQ (рис. 5 е) та S моделі HSI (рис. 5 ж).

6) Результат аналізу складової H моделі HSI (рис. 5 є) має найбільш перспективні результати, бо найкраще виділив три основні рівні зайнятості територій рослинністю.

Висновки. Було розроблено програмне забезпечення для аналізу інформативності кольорових складових кольорових моделей.

На прикладі зображення поля та кластерного аналізу зроблено висновки про перспективність використання додаткових кольорових систем, крім RGB, та доцільність вивчення впливу на аналіз зображення вибору кольорової моделі та певної складової залежно від очікуваного результату.

Найбільш перспективним для розв’язання задачі аналізу посівної площі та розрізнення рослинності з землею є кольорова модель HSI та особливо складова H, що відповідає за відтінок.



а) б)

в) г)

д) е)

є) ж)

Рис.унок 5 – Результати кластеризації зображень на основі різних складових різних кольорових моделей зображення

Бібліографічні посилання

1. Біла К. О., Лигун А. О., Шумейко О. О. Комп'ютерна графіка. Дніпропетровськ. 2010. С. 83.

2. Курочкін В. М. Система «ElfinTest» обробки моніторингу довкілля на основі кластеризації // Наукоємні технології. № 2 (26). 2015. С. 127–133.

3. Курочкін В. М. Аналіз неоднорідних текстур посівних площ на основі оцінки суміші розподілів // Наукоємні технології. № 4 (28). 2015. С. 305–310.

4. Harmeet K. K., Prabhpreet K. A Review: Color Models in Image Processing // Int. J. Computer Technology & Applications. Vol. 5 (2). 2014. P. 319–322.

5. Гонсалес Р'Вудс Р. Цифровая обработка изображений. Москва. 2005. 1072 с.

6. Rasras R. J., El Emary I. M. M., Skopin D. E. Developing a New Color model for Image Analysis and Processing // ComSIS. Vol. 4. No. 1. 2007. P. 43–56.

7. Nakajima N., Taguchi A. A novel color image processing scemein HSI color space with negative image processing // ISPACS. 2014. P. 29–33.

8. Sharma B., Nayyer R. Use and Analysis of Color Models in Image Processing // Food processing and Technology. Vol. 7. 2015. URL: http://dx.doi.org/10.4172/2157–7110.1000533.

9. Szabolcs S. Color Content Based Image Classification // 5th Slovakian Hungarian Joint Symposium of AMII. 2007. URL: http://uni–obuda.hu/conferences/sami2007/ 42_Sergyan.pdf.

10. Pass G., Zabith R. Histogram Refinement for Content Based Image Retrieval // IEEE Workshop on Applications of Computer Vision. 1996. P. 96–102.

Надійшла до редколегії 13.10.16.

УДК 519.688:556

О.П. Луценко, О.Г. Байбуз, А.О. Чорна

Дніпропетровський національний університет імені Олеся Гончара

ОСОБЛИВОСТІ ЗАСТОСУВАННЯ МЕТОДІВ ВИЯВЛЕННЯ РОЗЛАДНАНЬ У ПРОЦЕСІ ДОСЛІДЖЕННЯ ДАНИХ ЕКОЛОГІЧНОГО МОНІТОРИНГУ В РЕГІОНАХ З ТЕХНОГЕННИМ НАВАНТАЖЕННЯМ

Досліджено статистичні властивості даних моніторингу хімічних показників складу води річки Саксагань та хвостосховища Північного гірничо-збагачуваного комбінату. Запропоновано способи перетворення та обробки даних, а також модифікації та налаштування методів пошуку розладнання, які дозволяють ефективно отримати дані про розладнання процесів для подальших етапів дослідження.

Ключові слова*: нестаціонарні процеси, визначення розладнань, екологічний моніторинг.*

Исследованы статистические свойства данных мониторинга химических показателей состава воды реки Саксагань и хвостохранилища Северного горно-обогатительного комбината. Предложены способы преобразования и обработки данных, а также модификации и настройки методов поиска разладок, которые позволяют эффективно получать данные о разладках процессов для дальнейших этапов исследования.

Ключевые слова: *нестационарные процессы, выявление разладок, экологический мониторинг.*

The statistical properties of the chemical monitoring data of the Saksagan river and the tailings dam of the Northern Mining-enriched plant has been studied. The methods for data conversion and processing, as well as the modification and adjustment of change-point detection methods has been proposed. The methods and adjustments mentioned allow to effectively retrieve change-point data from processes of ecological monitoring for the further stages of the study.

Keywords: *non-stationary processes, change-point detection, ecology monitoring.*

Постановка проблеми. Аналіз часових рядів, утворених нестаціонарними процесами, був і залишається актуальним

напрямом досліджень з огляду на наявність великої кількості задач, що зводяться до роботи з часовими рядами. Задача екологічного моніторингу належить саме до такого класу задач. Існує значна кількість методів та підходів розв'язання задачі аналізу станів та прогнозування нестаціонарних процесів, серед яких можна виділити застосування методів лінічної та нелінійної регресії, нечіткої логіки, нейронних мереж, динамічного та еволюційного програмування.

Окремо слід зазначити напрям, який на сьогоднішній день є перспективним для застосування в галузі аналізу нестаціонарних процесів – ***методи виявлення розладнання***. Даний напрям дослідження представляє інтерес як засіб дослідження часових рядів через здатність методів реагувати на зміни статистичних характеристик ряду з мінімальною затримкою у часі, розмежовуючи ряд на ділянки з подібними статистичними властивостями.

В регіонах з підвищеним технологічним навантаженням такі зміни виникають при різкому впливі на навколишнє середовище, наприклад при екологічній катастрофі, при якій можлива швидка зміна контрольованих фізичних і хімічних показників. При обробці наукових спостережень, використовуючи методи виявлення розладнання, є можливим раннє виявлення відхилень у показниках моніторингу, локалізація джерел збурень і, як наслідок, своєчасне попередження наслідків цих збурень.

Розладнанням випадкового процесу називається стрибкоподібною зміною його властивостей, що відбувається в невідомий момент часу τ, або не відбувається взагалі. Завданням виявлення розладнання є встановлення факту розладнання, і якщо таке сталося, оцінювання моменту часу τ.

Виявлення розладнань у процесі екологічного моніторингу є складовою задачі контролю та мінімізації ризиків форс-мажорних обставин, так як дає можливість вчасно втрутитися у процес керування і зменшити фінансові втрати, пов'язані зі справдженням гіпотези про розладнання.

Аналіз літературних даних і постановка проблеми. Задача послідовного виявлення розладнання з моменту її формальної постановки в 1950-х роках отримала розвиток у працях вітчизняних і зарубіжних вчених. У працях [1; 2] були запропоновані непараметричні модифікації методів, що дозволяють виявляти розладнання при нестачі даних про статистичний розподіл процесу після моменту розладнання. Авторами [3; 4] було здійснено перехід до байєсівських методів аналізу часових рядів у задачі про розладнання. В попередніх працях авторів даної статті, зокрема [5; 6], було запропоновано моделі та

обчислювальні схеми ймовірнісного аналізу часових рядів, які дозволяли обчислювати ймовірність розладнання у часових рядах з множинними розладнаннями.

Якнайшвидше виявлення події розладнання і встановлення моменту часу τ в досліджуваному процесі є значущим чинником у процесі аналізу, а послідовні алгоритми, що використовуються для виявлення розладнань, повинні залишатися працездатними при будь-яких відхиленнях статистичних характеристик спостережуваних сигналів. При цьому слід зазначити, що властивості вхідних часових рядів, а також характер розладнань, які повинні бути виявлені в задачі екологічного моніторингу, обумовлюють виникнення специфічних підзадач, пов'язаних з перетвореннями вхідних послідовностей і налаштуванням методів таким чином, щоб забезпечити достатню якість виявлення. Ідентифікація і вирішення таких підзадач є необхідною складовою вирішення задачі ідентифікації розладнань.

Мета даної статті – на основі комплексного аналізу вхідних даних обрати найбільш придатні для проведення подальшого дослідження алгоритми виявлення розладнань та встановити їхні налаштування для використання у процесі аналізу рядів екологічного моніторингу.

Основний матеріал. Дані, що розглядаються в рамках дослідження – гідрохімічний склад вод районів з підвищеним технологічним навантаженням, зокрема дані хімічних показників складу води річки Саксагань (рис. 1) та хвостосховища Північного гірничо-збагачувального комбінату (рис. 2).

Перед побудовою моделей виявлення розладнань необхідно провести попередній аналіз цих даних для виявлення відмінних статистичних особливостей процесів. Далі наведено результати перевірки статистичних властивостей рядів, з урахуванням яких виконувалися подальші етапи дослідження.

Отримані коефіцієнти кореляції між гідрохімічними показниками вод р. Саксагань та хвостосховища наведено у табл. 1 і 2. За ними можна зробити висновок, що для обох водних джерел між усіма даними проявляється сильна або середня залежність. Тільки показник РН кислотність слабо корельований з іншими показниками. Це можна бачити і за графіком його рівнів.

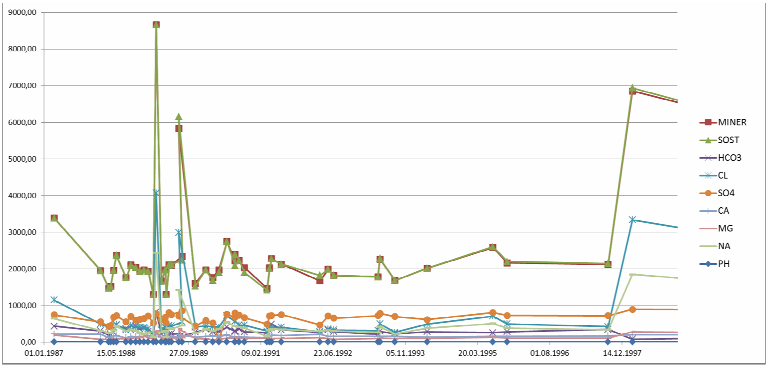


Рисунок 1 – Часові ряди даних річки Саксагань

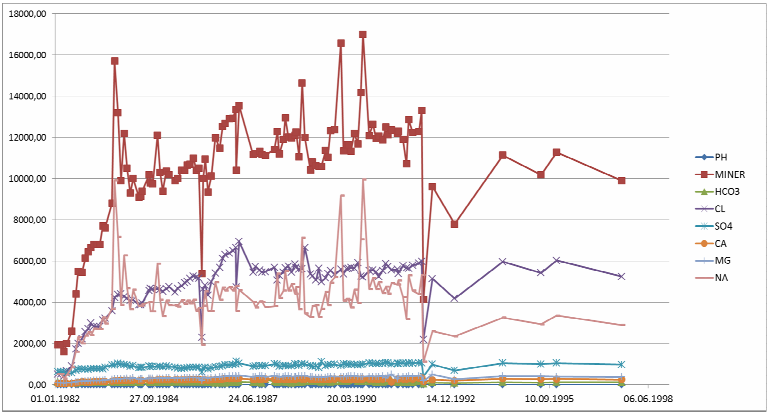


Рисунок 2 – Часові ряди даних хвостосховища Північного гірничо-збагачувального комбінату

Таблиця 1 – Коефіцієнти кореляції між даними р. Саксагань

	Mineral	PH	HCO3	SO4	Cl	Ca	Mg	Na
Mineral	1	-0,04569	0,380594	0,836438	0,829786	0,500989	0,575182	0,940915
PH	-0,04569	1	0,231659	0,019775	-0,26862	-0,17319	0,14119	-0,1378
HCO3	0,380594	0,231659	1	0,172162	0,161664	0,36426	0,13212	0,258212
SO4	0,836438	0,019775	0,172162	1	0,591462	0,40115	0,546197	0,735907
Cl	0,829786	-0,26862	161664	0,591462	1	0,338051	0,426466	0,881301
Ca	0,500989	-0,17319	0,36426	0,40115	0,338051	1	0,258497	0,418449
Mg	0,575182	0,14119	0,13212	0,546197	0,426466	0,258497	1	0,419196
Na	0,940915	-0,1378	0,258212	0,735907	0,881301	0,418449	0,419196	1

Таблиця 2 – Коефіцієнти кореляції між даними хвостосховища

	Mineral	PH	HCO3	SO4	Cl	Ca	Mg	Na
Mineral	1	0,242569	0,262598	0,740274	0,791098	0,776215	0,734273	0,886257
PH	0,242569	1	0,071485	0,104478	0,338202	0,310287	0,356376	0,206522
HCO3	0,262598	0,071485	1	0,292346	0,28468	0,291074	0,311122	0,167929
SO4	0,740274	0,104478	0,292346	1	0,702331	0,786697	0,731661	0,582591
Cl	0,791098	0,338202	0,28468	0,702331	1	0,883866	0,8854	0,516075
Ca	0,776215	0,310287	0,291074	0,786697	0,883866	1	0,822179	0,538863
Mg	0,734273	0,356376	0,311122	0,731661	0,8854	0,822179	1	0,492041
Na	0,886257	0,206522	0,167929	0,582591	0,516075	0,538863	0,492041	1

Перед виділенням тренда було перевірено дані на його наявність за допомогою критеріїв: ранговий критерій Спірмена і критерій, заснований на знаках різниці. Також використовувався критерій випадковості, заснований на серіальній кореляції (рис. 1.3), який будується за коефіцієнтами автокореляції. Аналіз даних на випадковість показав, що покажчики річки Саксагань є випадковими даними, в той час як у рядах хвостосховища виявлено зростаючий тренд.

Виділення тренда здійснювалося:

— методом лінійної регресії: $y(t) = a_0 + a_1 t + e_t$,

— методом нелінійної регресії – модифікована експоненціальна крива зростання: $y(t) = a_0 a_1^t + e_t + \gamma$.

У цих методах значення параметрів a_0 , a_1^t знаходять методом найменших квадратів, e_t – випадкова складова, γ – обчислюється методом трьох точок.

Для перевірки адекватності та значущості побудованих трендів використовувався дисперсійний аналіз. Вибір найкращого тренда проводився за байєсівським інформаційним критерієм Шварца – BIC.

Для даних хвостосховища усі моделі нелінійного тренда виявилися неадекватними. Деякі лінійні моделі тренда були неадекватними, але в них похибка незначна і не перевищує 5 %. Цим можна знехтувати, так як наявність точок розладнань ускладнює виділення тренда. По критерію BIC для усіх часових рядів хвостосховища лінійний тренд визнано кращим.

Також було перевірено два види моделей сезонності:

— в адитивній моделі сезонність виражається у вигляді абсолютної величини, яка додається або віднімається з середнього значення ряду для обліку сезонності;

— в мультиплікативній моделі сезонність виражається як відсоток від середнього рівня, на який множиться середнє значення для обліку сезонності при прогнозуванні.

Сезонність будувалася за допомогою гармонічного аналізу, використовуючи розклад у ряд Фур'є. Виявлення сезонності необхідно проводити по стаціонарному ряду. Тому для тих даних, де є трендова складова, вона попередньо була вилучена. Перевірка адекватності та значущості тренд-сезонної моделі проводилася за допомогою дисперсійного аналізу.

Для даних річки Саксагань після виділення сезонності і побудови тренд-сезонної моделі для усіх покажчиків, крім РН, модель виявилася неадекватною. Для даних хвостосховища ситуація є аналогічною.

Таким чином:

1. Числові ряди гідроекологічних показників, що розглядаються, характеризуються високими значеннями автокореляції та значною кореляцією між окремими показниками.

2. У вхідних даних сезонність відсутня.

3. Дані по р. Саксагань є стаціонарними за середнім значенням, в той час як числові ряди хвостосховища містять у собі зростаючий тренд, який ускладнює коректне виявлення розладнання.

Ключовою особливістю розглядуваних даних є наявність у них моментів часу, коли вони змінюють статистичні властивості. В даних, зібраних з річки Саксагань, спостерігаються єдиноразові викиди показників вгору, в той час як процеси, що протікають у хвостосховищі, утворюють ряди з локально направленими трендами на фоні глобального.

Для вирішення завдання представляється доцільним використання як послідовних методів виявлення розладнання, так і алгоритмів апостеріорного класу. Послідовні методи повинні забезпечити обробку новітніх даних, що надходять, в той час як апостеріорні – слугувати для побудови основної вибірки точок розладнання на вже існуючих даних у минулому, а також при наявності можливості одночасного застосування обох класів методів до однієї і тієї самої вибірки, бути контрольними методами.

Досліджувані процеси належать до класу нестаціонарних випадкових процесів з множинними (повторюваними) розладнаннями. При цьому характер спостережуваної при розладнанні зміни статистичних характеристик відрізняється, і статистичні

характеристики процесу після розладнання невідомі. У зв'язку з таким характером досліджуваного процесу, для подальшого дослідження більшою мірою підходять непараметричні методи, які зовсім не потребують інформації про розподіл процесу після розладнання, або методи з низькими вимогами до параметризації характеру розподілу після розладнання.

У зв'язку з тим, що розладнання в одному з досліджуваних процесів (р. Саксагань) являє собою викиди вгору на фоні незмінного середнього значення на ділянках без розладнання, для використовуваних методів бажана здатність виявлення змін статистичних характеристик (в даному випадку – середнього значення) як в обидві сторони, так і в одну сторону.

З урахуванням названих критеріїв, для вирішення задачі послідовного виявлення була задіяна непараметрична форма алгоритму кумулятивних сум [2].

На кожному кроці алгоритму накопичується сума:

$$S_t = \max(0,\, S_{t-1} + g_t)\,.$$

Сигнал про розладнання подається у момент часу:

$$\tau = \inf(t \geq 1 : S_t > b)\,,$$

де b – бар'єр чутливості методу.

Для випадку збільшення математичного очікування m:

$$g_t = z(t_i) - m_1 - k \quad при\ m_2 > m_1\,,$$

для випадку зменшення m:

$$g_t = -z(t_i) + m_1 + k \quad при\ m_2 < m_1\,,$$

де m_1, m_2 – математичне очікування величини y до і після розладнання відповідно,

$k \geq 0$ – поріг чутливості методу для відхилення від m_1.

Обрані обчислювальні схеми послідовного виявлення моменту розладнання не можуть бути застосовані у випадку нестаціонарних процесів з направленим характером змін середнього, так як сигнал про розладнання виникатиме на кожному кроці спостережень. При послідовному аналізі процесів з направленим характером змін середнього значення (вираженим трендом), необхідно здійснити таке перетворення часового ряду, при якому зберігалися б постійні середні значення. У якості таких перетворень авторами запропоновано такі:

1) обчислення ряду кінцевих різниць і знаходження розладнань його середньої величини. Оскільки стабільний тренд сам по собі характеризується зміною середнього значення спостережень, у даному випадку доцільно застосовувати АКС не до вихідної кривої

спостережень, а до її першої похідної за часом. Різка зміна значення першої похідної буде означати зміну напряму графіка часового ряду, тобто розладнання початкового процесу;

2) обчислення ряду значень нахилів апроксимуючих відрізків (рис. 2.3). Для кожної точки часового ряду знаходяться рівняння прямої, яка найкраще апроксимує задану кількість точок l часового ряду зліва від заданої точки. Позначимо: y_t – значення точки часового ряду, яку спостерігаємо; t – відповідний момент часу; k , b – коефіцієнти рівняння апроксимаційної прямої.

Досягнемо найкращої апроксимації за допомогою рішення такої системи (мінімізуємо сумарне квадратичне відхилення точок апроксимаційної прямої від кривої спостережень):

$$y_j = kt_j - b,$$

$$\sum_{j=i-l}^{j=i} (y_j - kt_j - b)^2 \to \min,$$

$$\sum_{j=i-l}^{i} y_j t_j = k \sum_{i=t-l}^{i} t_j^2 + b \sum_{j=i-l}^{i} t_j,$$

$$\sum_{j=i-l}^{i} y_j = k \sum_{j=i-l}^{i} t_j + nb.$$

Знайдемо коефіцієнт k, що є тангенціальним коефіцієнтом рівняння прямої. Визначивши кути arctg(k) для усіх точок часового ряду, отримаємо ряд нахилів апроксимуючих прямих до горизонтальної осі. Знаходячи розладнання даного ряду, отримуємо набір точок розладнань для часового ряду з вираженим трендом.

В якості методів послідовного виявлення було використано апостеріорний тест Бродського – Дарховського [1; 7]:

$$Y(\tau) = \frac{\tau}{N}\left(1 - \frac{\tau}{N}\right)\left(\frac{1}{\tau}\sum_{1}^{\tau} y_i - \frac{1}{N-\tau}\sum_{\tau+1}^{N} y_i\right),$$

$$G = \max_{1 \le \tau \le N-1} |Y(\tau)|,$$

$$G < h \Rightarrow \textit{дефекта немає},$$

$$G \ge h \Rightarrow \textit{дефект},$$

$$T(\tau) = Y(\tau + [\varepsilon N]) - Y(\tau),\ 0 < \varepsilon < \frac{\delta}{4}.$$

$$\tau_1 = \begin{cases} \min A_1, A_1 \neq \varnothing \\ N, A_1 = \varnothing \end{cases},$$

$$A_1 = \{\tau \geq 1 : sign(T(\tau)) \neq sign(T(\tau+1))\},$$

$$\tau_i = \begin{cases} \min A_i, A_i \neq \varnothing \\ N, A_i = \varnothing \end{cases},$$

$$A_i = \{\tau \geq \tau_{i+1} + [\delta N / 2] : sign(T(\tau)) \neq sign(T(\tau+1))\},$$

$$i = 2..\tilde{k},$$

$$\tilde{k} = \min\{s : \tau_s = N\} - 1.$$

Даний алгоритм має важливу для поставленої задачі особливість – здатність виявляти повторювані розладнання на часовому ряді. В той же час більш класичні методи, зокрема тест Манна – Уітні [7], виявляють лише одне розладнання на інтервалі:

$$u_{ik} = \begin{cases} 1, y_i \geq y_k \\ 0, y_i < y_k \end{cases},$$

$$G(\tau) = \frac{\sum_{k=\tau+1}^{N} \sum_{i=1}^{n} u_{ik}}{\tau(N-\tau)},$$

$$\tau_0 = \arg \min_{[\alpha N] \leq \tau \leq [N-\alpha N]} G(\tau), \mu_1 < \mu_2,$$

$$\tau_0 = \arg \max_{[\alpha N] \leq \tau \leq [N-\alpha N]} G(\tau), \mu_1 > \mu_2,$$

де τ_0 – момент розладнання.

Слід, однак, зазначити, що алгоритми, засновані на виявленні розладнання як локального мінімума або максимума деякої допоміжної статистики, можуть бути налаштовані на виявлення множинних розладнань. Для цього в рамках вирішуваної задачі був застосований рекурсивний порядок виклику алгоритму на ділянках зліва і справа від розладнання: (0; τ_0), (τ_0+1, N). Умови завершення рекурсії: на ділянці відсутній локальний максимум $G(\tau)$ або локальний максимум $G(\tau)$ не перевищує встановлений бар'єр.

Приклад результату застосування алгоритмів до часового ряду наведено на рис. 3.

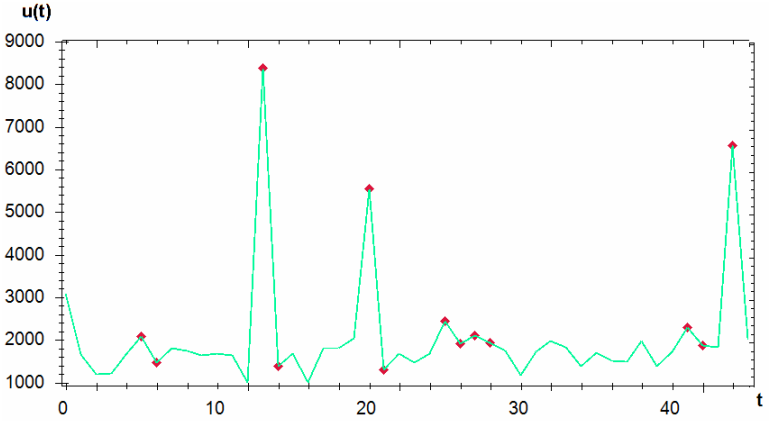


Рисунок 3 – Точки розладнання на часовому ряді

Уточнення результатів виявлення розладнань за допомогою апостеріорних алгоритмів проводилося таким чином: після виявлення розладнання послідовним алгоритмом у точці τ_S апостеріорний алгоритм запускався на ділянці в околі точки розладнання: $(\tau_{si}-\alpha;\tau_{si}+\alpha)$, де α – величина допуску, яка налаштовується. За відсутності сигналу про розладнання від апостеріорного алгоритму сигнал про розладнання вважався помилковим. Щоб виявити кількість моментів розладнання, що були пропущені послідовним алгоритмом, послідовний алгоритм запускався на інтервалі часу в минулому, після чого виявлялися точки, знайдені апостеріорним алгоритмом, в околі яких $(\tau_{aj}-\beta;\tau_{aj}+\beta)$ відсутні точки розладнання τ_{si}, що виявлені послідовно.

Вказані дії дозволили виявити ймовірність помилок першого і другого роду при застосуванні кожного набору налаштувань послідовного алгоритму і обрати налаштування алгоритмів згідно з цими критеріями.

Висновки. Таким чином, отримані налаштування та запропоновані модифікації методів виявлення розладнань, які дозволяють адаптувати методи до застосування в предметній галузі екологічного моніторингу.

Запропоновані методи усунення тренда, придатні для використання при послідовному виявленні розладнань. Дістала подальшого розвитку методика апостеріорного виявлення множинних розладнань. Запропоновано спосіб уточнення результатів послідовного виявлення

множинних розладнань за допомогою апостеріорних алгоритмів.

Отримані результати закладають основу для розширення області застосування підходів, запропонованих у [5; 6], на нові класи процесів, що характеризуються складнощами у процесі виявлення розладнань.

Бібліографічні посилання

1. Brodsky B. E., Darkhovsky B. S. Nonparametric Methods in Change-Point Problems. Dordrecht. Kluver Academic Publishings. 1993. 210 p.

2. Nadler J., Robbins N. B. Some characteristics of Page's twosided procedure for detecting a change in a location parameter. // Ann. Math. Statist. 1971. Vol. 42. N 2. P. 538–551.

3. Adams R. P., McKay D. J. C. Bayesian Online Changepoint Detection // arXiv preprint. 2007. URL: http://arxiv.org/pdf/0710.3742v1.pdf (дата звернення: 28.09.2016).

4. Adams R. P., Murray I., McKay D. J. C. Nonparametric Bayesian density modeling with Gaussian processes // arXiv preprint. 2009. URL: https://arxiv.org/pdf/0912.4896.pdf (дата звернення: 28.09.2016)

5. Lutsenko O., Baybuz O. Model of probabilistic assessment of trend stability at financial market // Eastern-European Journal of Enterprise Technologies. 2013. Vol. 6. N 3 (66). P. 50–54.

6. Луценко О. П., Байбуз О. Г. Оцінка функції ризику розладнання процесу коливань валютних курсів із застосуванням байєсівської оцінки параметра функції умовного розподілу // The European Scientific and Practical Congress «Global scientific unity 2014» (Prague (Czech Republic) 26–27 September 2014). Prague. 2014. С. 126–132.

7. Бродский Б. Е., Дарховский Б. С. Асимптотический анализ некоторых оценок в апостериорной задаче о разладке. // Теория вероятностей и ее применения. 1990. Т. 35. № 3. С. 551–557.

Надійшла до редколегії 01.10.2016.

УДК 519.254:519.233.3:612.143

О.М. Мацуга[1], І.В. Дроздова[2], А.К. Акімова[1]
[1]Дніпропетровський національний університет імені Олеся Гончара
[2]Державна установа «Український держаний науково-дослідний інститут медико-соціальних проблем інвалідності»

ТЕХНОЛОГІЯ ПОРІВНЯННЯ РЕЗУЛЬТАТІВ ДВОХ ДОБОВИХ МОНІТОРУВАНЬ АРТЕРІАЛЬНОГО ТИСКУ ПАЦІЄНТА

Розроблено обчислювальну технологію порівняння результатів двох добових моніторувань артеріального тиску. Створено програмне забезпечення, в якому її реалізовано. Виконано практичну апробацію технології на реальних даних, яка засвідчила її ефективність.

Ключові слова: *добове моніторування артеріального тиску, обчислювальна технологія, функція розподілу, двовибірковий критерій Колмогорова, критерій Вілкоксона.*

Разработана вычислительная технология сравнения результатов двух суточных мониторирований артериального давления. Создано программное обеспечение, в котором она реализована. Проведена практическая апробация технологии на реальных данных, которая подтвердила её эффективность.

Ключевые слова: *суточное мониторирование артериального давления, вычислительная технология, функция распределения, двувыборочный критерий Колмогорова, критерий Вилкоксона.*

Computer-oriented technology for results comparison of two daily blood pressure monitorings was developed. Software in which it was implemented was created. Testing results on real data corroborated technology efficiency.

Keywords: *daily blood pressure monitoring, computer-oriented technology, distribution function, two-sample Kolmogorov test, Wilcoxon test.*

Постановка проблеми. Добове моніторування артеріального тиску (ДМАТ) є сучасною та ефективною методикою, що використовується для виявлення і оцінки ефективності лікування артеріальної гіпертензії. На даний час розроблено методи і програмні засоби оперативного аналізу результатів ДМАТ для діагностування стану пацієнта. Проте подальший контроль за станом хворого та оцінка ефективності його лікування потребують проведення повторного ДМАТ і порівняння його результатів з попередніми. Як правило, при

цьому лікарі обмежуються співставленням значень певних показників моніторування, відмінності у яких можуть бути несуттєвими і обумовленими випадковістю. Подібне порівняння не можна вважати прийнятним. Дана робота націлена на розробку більш ефективної технології порівняння результатів двох ДМАТ пацієнта.

Аналіз останніх досліджень і публікацій. Аналіз наявних публікацій свідчить, що розроблено методи і програмні засоби оперативного аналізу результатів ДМАТ для діагностування стану пацієнта [1–4]. Про наявність обчислювальної технології порівняння результатів двох ДМАТ авторам не відомо.

Постановка задачі. Пацієнту двічі проведено ДМАТ. Результати першого ДМАТ представлено масивом вигляду

$$\left\{x_{1,t}^{(1)}, x_{2,t}^{(1)}, x_{3,t}^{(1)}, x_{4,t}^{(1)}, x_{5,t}^{(1)}; t=\overline{1, N_1}\right\},$$

результати другого – масивом

$$\left\{x_{1,t}^{(2)}, x_{2,t}^{(2)}, x_{3,t}^{(2)}, x_{4,t}^{(2)}, x_{5,t}^{(2)}; t=\overline{1, N_2}\right\},$$

де $x_{1,t}^{(1/2)}$ – значення систолічного артеріального тиску (САТ) під час 1-го або 2-го моніторування, заміряне в t-й момент доби;

$x_{2,t}^{(1/2)}$ – величина діастолічного артеріального тиску (ДАТ);

$x_{3,t}^{(1/2)}$ – частота серцевих скорочень (ЧСС);

$x_{4,t}^{(1/2)} = 0,5\left(x_{1,t}^{(1/2)} + x_{2,t}^{(1/2)}\right)$ – середній артеріальний тиск (СрАТ);

$x_{5,t}^{(1/2)} = x_{1,t}^{(1/2)} - x_{2,t}^{(1/2)}$ – пульсовий артеріальний тиск (ПАТ);

N_1 – кількість добових замірів під час 1-го моніторування;

N_2 – кількість добових замірів під час 2-го моніторування.

Слід зазначити, що перші три показники безпосередньо фіксуються під час моніторування, останні два є розрахунковими.

Необхідно:

1. Розробити обчислювальну технологію для порівняння результатів 1-го та 2-го ДМАТ пацієнта.
2. Програмно реалізувати розроблену технологію.
3. Апробувати розроблену технологію на реальних даних державної установи «Український державний науково-дослідний інститут медико-соціальних проблем інвалідності».

Основний матеріал. Показники ДМАТ, які підлягають обробці, не є постійними протягом доби. Вони коливаються, по-перше, внаслідок фізіологічних чинників, наприклад, вночі артеріальний тиск завжди

знижується, а по-друге, через випадкові зовнішні фактори: стреси, зміни погоди тощо. Зважаючи на наявність випадкового впливу, показники ДМАТ можна розглядати як випадкові величини і говорити про розподіл їх ймовірностей. Тоді, щоб перевірити, чи змінилися результати ДМАТ пацієнта після повторного обстеження, доцільно перевірити, чи змінилися функції розподілу показників.

З огляду на ці міркування пропонуємо обчислювальну технологію порівняння результатів двох ДМАТ пацієнта, яка базується на порівнянні функцій розподілу усіх показників ДМАТ.

Порівняння функцій розподілу показника x_j, $j=\overline{1,5}$ пропонуємо проводити на такою схемою:

1. Сформувати варіаційні ряди за результатами спостережень показника x_j під час 1-го та 2-го моніторувань і одержати

– $x_{j,1}'^{(s)}, x_{j,2}'^{(s)}, \ldots, x_{j,n_s}'^{(s)}$ – варіанти, тобто унікальні значення показника x_j під час s-го моніторування, розташовані за зростанням; n_s – кількість цих значень;

– $f_{j,1}^{(s)}, f_{j,2}^{(s)}, \ldots, f_{j,n_s}^{(s)}$ – частоти варіант, для яких буде справедливим $\sum_{h=1}^{n_s} f_{j,h}^{(s)} = N_s$;

– $p_{j,1}^{(s)}, p_{j,2}^{(s)}, \ldots, p_{j,n_s}^{(s)}$ – відносні частоти, розраховані як $p_{j,h}^{(s)} = \dfrac{f_{j,h}^{(s)}}{N_s}$, $j=\overline{1,n_s}$.

2. Побудувати емпіричні функції розподілу [5] показника x_j за даними 1-го та 2-го моніторувань:

$$F_s\left(x_j\right) = \begin{cases} 0, & x_j < x_{j,1}'^{(s)}, \\ \sum_{h=0}^{i} p_{j,h}^{(s)}, & x_{j,i}'^{(s)} \le x_j < x_{j,i-1}'^{(s)}, \\ 1, & x_j \ge x_{j,n_s}'^{(s)}, \end{cases}$$

де $s=1,2$ – номер моніторування.

3. Порівняти функції розподілу показника під час 1-го та 2-го моніторувань за двовибірковим критерієм Колмогорова [5]. З цією метою розрахувати статистику критерію

$$z = \sqrt{\frac{N_1 N_2}{N_1 + N_2}} D_{N_1,N_2},$$

де

$$D_{N_1,N_2} = \max\left\{D^-_{N_1,N_2}, D^+_{N_1,N_2}\right\};$$

$$D^-_{N_1,N_2} = \max_{h=\overline{1,n_2}}\left|F_1\left(x'^{(2)}_h\right) - F_2\left(x'^{(2)}_h\right)\right|;$$

$$D^+_{N_1,N_2} = \max_{h=\overline{1,n_1}}\left|F_1\left(x'^{(1)}_h\right) - F_2\left(x'^{(1)}_h\right)\right|.$$

За великих обсягів даних статистика z розподілена за законом Колмогорова. Тому потрібно обчислити значення функції Колмогорова

$$K(z) = 1 + 2\sum_{k=1}^{\infty}(-1)^k e^{-2k^2 z^2}.$$

Якщо при заданому рівні значущості α справедлива нерівність

$$p = 1 - K(z) < \alpha,$$

то можна стверджувати, що функція розподілу показника x_j під час 2-го моніторування суттєво відрізняється від тієї, що мала місце під час 1-го. У такому разі перейти до пункту 4. В іншому випадку слід вважати, що розподіл показника не змінився.

4. Перевірити, чи обумовлена відмінність у функціях розподілу показника зсувом і, якщо так, то функція розподілу за 2-го моніторування зсунена вправо чи вліво відносно функції розподілу за 1-го моніторування. З цією метою можна застосувати ранговий критерій зсуву, наприклад, Вілкоксона [5]. Для цього сформувати з послідовностей $\left\{x^{(1)}_{j,t}; t = \overline{1, N_1}\right\}$ та $\left\{x^{(2)}_{j,t}; t = \overline{1, N_2}\right\}$ загальну послідовність довжиною $N_1 + N_2$

$$\left\{x^{(1)}_{j,1}, \ldots x^{(1)}_{j,N_1}, x^{(2)}_{j,1}, \ldots x^{(2)}_{j,N_2}\right\},$$

елементам якої приписати ранги

$$\left\{r_1, \ldots r_{N_1}, r_{N_1+1}, \ldots r_{N_1+N_2}\right\}.$$

Тоді розрахувати статистику критерію

$$W = \sum_{t=1}^{N_1} r_i$$

та стандартизовану статистику

$$u = \frac{W - E\{W\}}{\sqrt{D\{W\}}},$$

де

$$E\{W\} = \frac{1}{2} N_1 \left(N_1 + N_2 + 1\right);$$

$$D\{W\} = \frac{1}{12} N_1 N_2 \left(N_1 + N_2 + 1\right).$$

Якщо $|u| \le u_{1-\alpha/2}$ ($u_{1-\alpha/2}$ – квантиль стандартного нормального розподілу), то функція розподілу показника за 2-го моніторування не зсунена відносно функції розподілу показника за 1-го моніторування. Коли $u > u_{1-\alpha/2}$ має місце зсув вліво, інакше – вправо.

Запропоновану обчислювальну технологію було реалізовано у програмному забезпеченні ArtHyper, яке написано мовою C# у середовищі Microsoft Visual Studio 2012.

За його допомогою здійснено практичну апробацію запропонованої технології на реальних даних, які було зібрано у такий спосіб. У державній установі «Український державний науково-дослідний інститут медико-соціальних проблем інвалідності» хворим на артеріальну гіпертензію 2-ї та 3-ї стадій після корекції їх стану проведено ДМАТ на апараті АВРМ-01 (Meditech, Угорщина). Через рік пацієнтам повторно проведено ДМАТ на тому самому апараті. При цьому частина хворих протягом року дотримувалася призначеного лікування, інші пацієнти відмовилися від нього. Постає питання: чи змінився стан хворих через рік і чи було це лікування ефективним?

В якості прикладу нижче наведено результати застосування розробленої технології для двох пацієнтів – одного, що дотримувався призначеного лікування, та другого, який знехтував ним. Під час застосування усіх критеріїв використано $\alpha = 0{,}05$.

Для пацієнта, який дотримувався призначеного лікування, емпіричні функції розподілу показників САТ, ДАТ, СрАТ, ПАТ та ЧСС, побудовані за даними 1-го та повторного ДМАТ, подано на рис. 1.

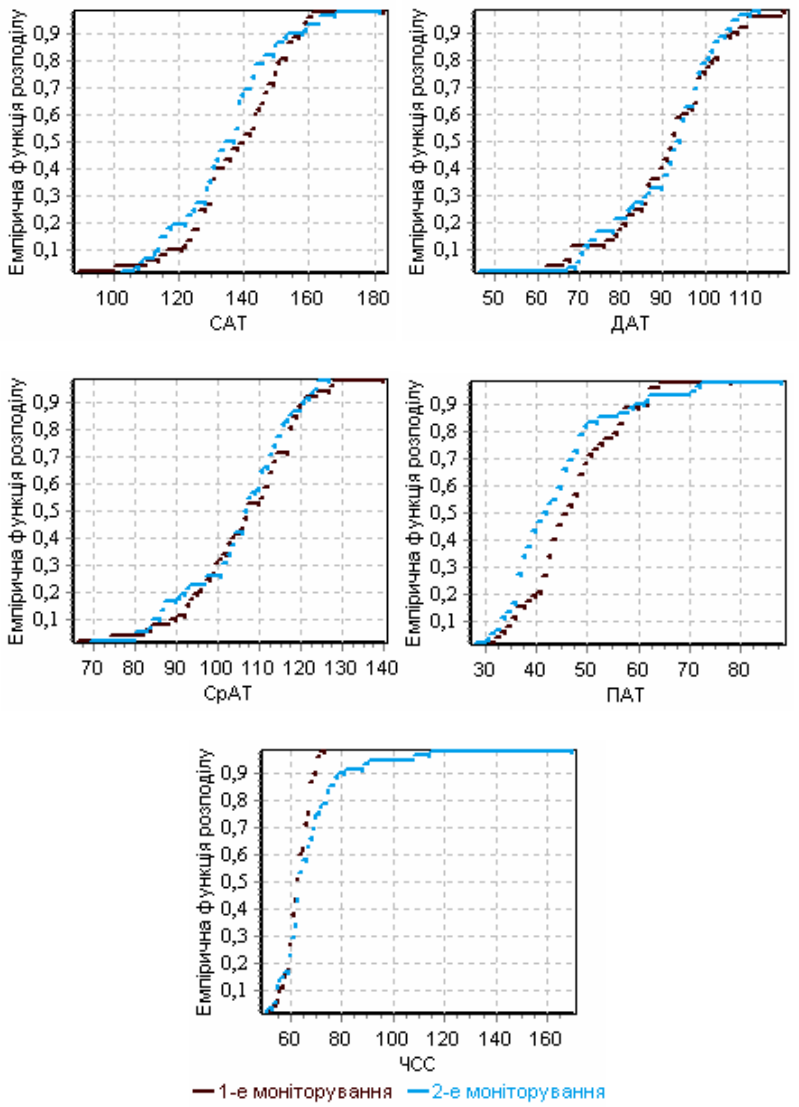


Рисунок 1 – Емпіричні функції розподілу показників 1-го та 2-го ДМАТ для пацієнта 1

Згідно з результатами двовибіркового критерію Колмогорова функції розподілу ПАТ для двох моніторувань різняться суттєво, для інших показників вони рівні (табл. 1). За критерієм Вілкоксона функція розподілу ПАТ, одержана під час 2-го моніторування, зсунена вліво відносно тієї, що мала місце під час першого обстеження пацієнта. Це свідчить про те, що рівень ПАТ через рік знизився.

Таблиця 1 – Результати застосування двохвибіркового критерію Колмогорова для пацієнта 1

Показник	D_{N_1,N_2}	z	$K(z)$	p	Висновок щодо рівності функцій розподілу показника під час 1-го та 2-го моніторувань
САТ	0,195	1,041	0,771	0,229	функції розподілу рівні
ДАТ	0,085	0,454	0,014	0,986	функції розподілу рівні
СрАТ	0,122	0,651	0,209	0,791	функції розподілу рівні
ПАТ	0,260	1,391	0,958	0,042	функції розподілу відрізняються
ЧСС	0,212	1,134	0,847	0,153	функції розподілу рівні

Таблиця 2 – Результати застосування критерію Вілкоксона для пацієнта 1 ($\boldsymbol{u_{1-\alpha/2} = 1,96}$)

Показник	W	u	p	Висновок щодо зсуву функції розподілу показника під час 2-го моніторування
САТ	3327,5	1,422	0,155	зсуву немає
ДАТ	3067	-0,039	0,969	зсуву немає
СрАТ	3169,5	0,536	0,592	зсуву немає
ПАТ	3463	2,183	0,029	зсув вліво
ЧСС	2801	-1,532	0,126	зсуву немає

Отже, для даного пацієнта розподіл більшості показників ДМАТ через рік не змінився, тобто завдяки лікуванню рівень артеріального тиску та частоти серцевих скорочень хворого залишилися на стабільному рівні. Трохи зменшився рівень ПАТ, але в даному випадку це є позитивним фактором.

Для пацієнта, що відмовився від лікування, емпіричні функції розподілу показників САТ, ДАТ, СрАТ, ПАТ та ЧСС, побудовані за даними 1-го та повторного ДМАТ, подано на рис. 2.

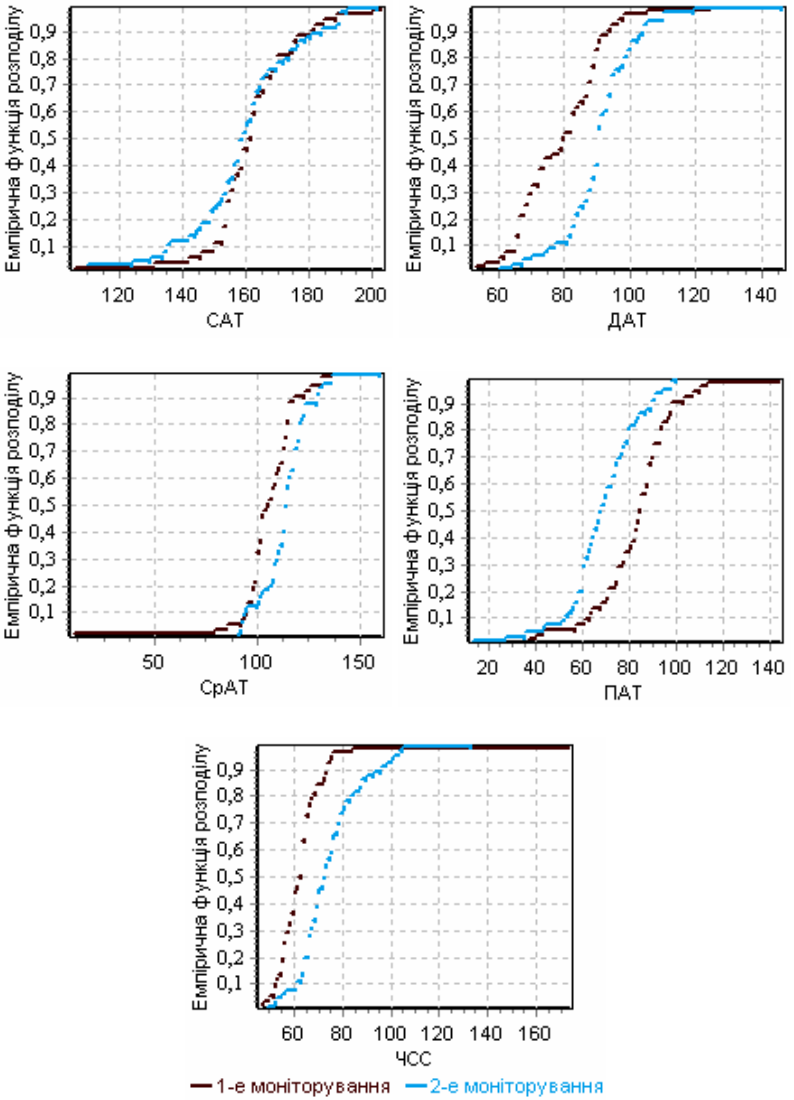


Рисунок 2 – Емпіричні функції розподілу показників 1-го та 2-го ДМАТ для пацієнта 2

Результати їх порівняння за двовибірковим критерієм Колмогорова свідчать, що розподіл САТ через рік залишився незмінним, розподіл чотирьох інших показників змінився (табл. 3). При цьому за результатами застосування критерію Вілкоксона (табл. 4) можна сказати, що функції розподілу ДАТ, СрАТ та ЧСС через рік виявилися зсуненими вправо відносно тих, що мали місце на початку. Отже, рівень цих показників через рік збільшився. Функція розподілу ПАТ виявилася зсуненою вліво, що є логічним у даному випадку.

Таблиця 3 – Результати застосування двовибіркового критерію Колмогорова для пацієнта 2

Показник	D_{N_1,N_2}	z	$K(z)$	p	Висновок щодо рівності функцій розподілу показника під час 1-го та 2-го моніторувань
САТ	0,162	0,868	0,562	0,438	функції розподілу рівні
ДАТ	0,427	2,295	1	0	функції розподілу відрізняються
СрАТ	0,342	1,840	0,998	0,002	функції розподілу відрізняються
ПАТ	0,454	2,439	1	0	функції розподілу відрізняються
ЧСС	0,531	2,853	1	0	функції розподілу відрізняються

Таблиця 4 – Результати застосування критерію Вілкоксона для пацієнта 2 ($\boldsymbol{u_{1-\alpha/2} = 1,96}$)

Показник	W	u	p	Висновок щодо зсуву функції розподілу показника під час 2-го моніторування
САТ	3234,5	0,913	0,361	зсуву немає
ДАТ	2153,5	-5,016	0	зсув вправо
СрАТ	2440	-3,445	0,001	зсув вправо
ПАТ	3928	4,717	0	зсув вліво
ЧСС	2049	-5,589	0	зсув вправо

Таким чином, за рік стан пацієнта, що недотримувався призначеного лікування, дещо погіршився, в нього зросли рівні діастолічного артеріального тиску та частоти серцевих скорочень.

Висновки про стан пацієнтів, одержані внаслідок застосування запропонованої технології порівняння результатів 1-го та потворного ДМАТ, підтверджуються додатковими медичними обстеженнями цих пацієнтів, є коректними і обґрунтованими. Це дозволяє говорити, що запропонована технологія може бути ефективною під час оцінки результативності реабілітації і лікування.

Висновки. Розроблено обчислювальну технологію порівняння результатів двох ДМАТ пацієнта, яку реалізовано у програмному забезпеченні ArtHyper. Технологія базується на порівнянні функцій розподілів показників ДМАТ за допомогою двовибіркового критерію Колмогорова та критерію Вілкоксона. Здійснено її практичну апробацію на реальних медичних даних, яка засвідчила ефективність технології для виявлення змін у результатах повторного моніторування та можливість її застосування для оцінки результативності реабілітації і лікування.

Бібліографічні посилання

1. Дзяк Г. В., Колесник Т. В., Погорецкий Ю. Н. Суточное мониторирование артериального давления. Днепропетровск. 2005. 200 с.

2. Хачапуридзе Т. Н. Моделирование мониторинга сердечно-сосудистой системы // Актуальні проблеми автоматизації та інформаційних технологій: зб. наук. праць. Дніпропетровськ. 2003. Т. 7. С. 142–149.

3. Дереза А. Ю., Приставка П. О. Кусково-марківська модель процесу зміни артеріального тиску за часом // Актуальні проблеми автоматизації та інформаційних технологій: зб. наук. праць. Дніпропетровськ. 2005. Т. 9. С. 3–12.

4. Мацуга О. М., Дереза А. Ю. Реалізація сумішей розподілів в автоматизованій системі ViStA Med // Актуальні проблеми автоматизації та інформаційних технологій: зб. наук. праць. Дніпропетровськ. 2006. Т. 10. С. 15–25.

5. Бабак В. П., Білецький А. Я., Приставка О. П. Статистична обробка даних. Київ. 2001. 388 с.

Надійшла до редколегії 30.06.2016.

УДК 519.254:519.652(045)

П.О. Приставка

Національний авіаційний університет

ПОШУК ОСОБЛИВИХ ТОЧОК ЦИФРОВОГО ЗОБРАЖЕННЯ ТА РОЗПІЗНАВАННЯ ОБ`ЄКТІВ НА ОСНОВІ СПЛАЙН-МОДЕЛІ

Запропоновано новий метод розпізнавання об'єктів на цифрових зображеннях. Метод базується на пошуку особливих точок, виходячи зі сплайн-моделі зображення.

Ключові слова: *сплайн, цифрова обробка зображень, розпізнавання.*

Предложен новый метод распознавания объектов на цифровых изображениях. Метод основывается на поиске особых точек, исходя из сплайн-модели изображения.

Ключевые слова: *сплайн, цифровая обработка изображений, распознавание.*

A new method for object recognition in digital images. The method is based on finding the singular points based on the spline model image.

Keywords: *spline, digital image processing, recognition***.**

Постановка проблеми. Розпізнавання об'єктів на цифрових зображеннях (ЦЗ) широко представлене в наукових та прикладних дослідженнях як невід'ємна складова окремої галузі цифрової обробки зображень, такої як «комп'ютерне бачення», що достатньо стрімко розвивається в останні десятиріччя. Інтерес до подібних методів розпізнавання виявляють розробники технічних систем та програмного забезпечення в галузі охоронних систем, систем різноманітного моніторингу, систем навігації по оптичному каналу, систем спеціального та військового призначення тощо.

Проте, незважаючи на очевидний прогрес розвитку інформаційних технологій обробки ЦЗ та відео на основі методів розпізнавання, актуальним залишається вирішення проблеми підвищення адекватності розпізнавання та швидкодії такої операції, бажано у режимі реального часу. В умовах критичності часу обробки невиправданим є застосування обчислювально обтяжливих підходів до розпізнавання та класифікації, наприклад, методів на основі головних компонент [1], дискримінантного аналізу [2] чи методу активних форм [3].

Найчастіше використання мають методи, що базуються на пошуку особливостей об'єктів на зображеннях, які інваріантні до лінійних перетворень та масштабу, такі як запатентований метод SIFT (Scale-Invariant Feature Transform), опублікований Девісом Лоуі в 1999 р. [4], або метод SURF (Speeded Up Robust Feature), розроблений в 2006 р. Гербертом Беєм [5] під впливом алгоритму SIFT. Проте, варто зазначити, що в основу методів пошуку особливих точок покладено модель зображення, згладженого гауссіаном. Авторська модель зображення [6] на основі лінійної комбінації B-сплайнів, близької до інтерполяційних у середньому, маючи усі переваги гауссової моделі (згладжування, маштабованість, аналітичний вигляд) показує більш низьку обчислювальну складність, а, отже, може мати перевагу при розробці методів розпізнавання на її основі, що реалізуються в програмному забезпеченні on-line обробки цифрових зображень та відео.

Аналіз публікацій та постановка задачі. Доволі розлогий аналіз методів на основі пошуку особливих точок зображення наведено Б. Г. Кухаренко в оглядовій роботі [7]. Але щоб не повторювати заміст даної корисної публікації, варто зауважити, що чимала кількість вагомих результатів з класифікації та розпізнавання в режимі реального часу залишається поза відкритим друком або має суто демонстраційний характер, як, наприклад, відома робота [8].

В основі сучасних ефективних та швидкодіючих методів розпізнавання, інваріантних відносно обертань та зміни масштабу, покладено аналіз часткових похідних зображення першого та другого порядку. Проте, через високу чутливість до зашумлення похідних, їх застосовують після згладжування зображень, переважно фільтрами на основі функції Гаусса. Отже, нехай у неперервній моделі двовимірного зображення $p(t,q)$ в якості функції імпульсного виклику використовується функція Гаусса [7]

$$L(t,q,g) = \int_{(\xi,\eta)} p(\xi,\eta) G(t-\xi, q-\eta, g)\, d\xi\, d\eta\,, \qquad (1)$$

де $G(t,q,g)$ – гауссова функція з параметризацією $g = \sigma^2$:

$$G(t,q,g) = \frac{1}{2\pi g} \exp\left(-\frac{t^2+q^2}{2g} \right),$$

тобто варіація $g = \sigma^2$ цього ядра є масштабним параметром [9]. Функції (1) $L(t,q,g)$ представляють ту саму інформацію, що й вихідне зображення $p(t,q)$, але не різних рівнях масштабу. Сімейство функцій

(1) у просторі масштаб – положення є розв'язком лінійного рівняння дифузії

$$\frac{\partial L}{\partial g} = \frac{1}{2}\nabla^2 L ,$$

з початковою умовою $L(t,q,0) = p(t,q)$. Аналогічно формулі

$$\nabla^2 \left(G\left(t,q,g\right) * p(t,q)\right) = \left(\nabla^2 G\left(t,q,g\right)\right) * p(t,q) ,$$

усі похідні (1) при довільному масштабі g можуть бути обчислені або безпосередньо диференціюванням, або обчисленням згортки зображення та відовідних часткових похідних функції Гаусса:

$$L_{t^\alpha q^\beta}\left(t,q,g\right) = \frac{\partial^{\alpha+\beta} L\left(t,q,g\right)}{\partial t^\alpha q^\beta} = \left(\frac{\partial^{\alpha+\beta} G\left(t,q,g\right)}{\partial t^\alpha q^\beta} \right) * p\left(t,q\right), \ \alpha,\beta = 1,2,\ldots \quad (2)$$

Цей спосіб визначення гауссових похідних робить спочатку погано визначену задачу диференціювання сімейства (1) добре визначеною і близько позв'язаною з узагальненими функціями.

Гессіан H – це матриця розмірності 2x2, яка виникає при другому члені розкладу Тейлора інтенсивності зображення $p(t,q)$:

$$p(t+u,q+v) - p(t,q) \approx \frac{\partial p}{\partial t}u + \frac{\partial p}{\partial q}v = s\,\nabla I ,$$

де $s = \begin{bmatrix} u & v \end{bmatrix}$ – вектор зсуву і

$$\nabla I = \begin{bmatrix} \dfrac{\partial p(t,q)}{\partial t} & \dfrac{\partial f(t,q)}{\partial q} \end{bmatrix}.$$

Тоді гессіан H виражається через похідні другого порядку (2) так:

$$\mathrm{H}\left(t,q,g\right) = \begin{bmatrix} L_{tt}\left(t,q,g\right) & L_{tq}\left(t,q,g\right) \\ L_{tq}\left(t,q,g\right) & L_{qq}\left(t,q,g\right) \end{bmatrix}.$$

Набір гауссових похідних для сімейства (2) у просторі масштаб – положення аж до порядку r в даній точці зображення і при заданому масштабі називається r-джетом і відповідає обрізаному розкладу Тейлора для локально згладженого фрагменту зображення [10]. Ці похідні разом описують базові види особливостей у просторі масштаб – положення та компактно представляють локальну структуру зображення. Для $r = 2$ при вибраному масштабі 2-джет містить гауссові похідні

$$\left(L\,,L\,,L\,,L\,,L\right). \quad (3)$$

З п'яти компонент 2-джета може бути сконструйовано чотири

диференціальних інваріанти відносно локальних обертань – магнітуда градієнта $|\nabla L|$, лапласіан $\nabla^2 L$, детермінант Гессіана $\det \mathrm{H}$ і кривизна кривої масштабування $\tilde{k}$:

$$|\nabla L| = L_t^2 + L_q^2, \tag{4}$$

$$\nabla^2 L = L_{tt} + L_{qq}, \tag{5}$$

$$\det \mathrm{H} = L_{tt} L_{qq} - L_{tq}^2, \tag{6}$$

$$\tilde{k} = L_t^2 L_{qq} + L_q^2 L_{tt} - 2 L_t L_q L_{tq}. \tag{7}$$

Детектор єдиного масштабу для знаходження структур типу крапель, який реагує на яскраві і темні структури, схожі на краплі, може базуватися на мінімумі і максимумі лапласіана $\nabla^2 L$. Афінно-коваріантний детектор структур типу крапель, який також реагує на сідла, може бути представлений як максимум і мінімум детермінанта гессіана $\det \mathrm{H}$. Прямолінійний і афінно-коваріантний детектор кутів може бути представлений як максимум і мінімум кривої рівня масштабування $\tilde{k}$.

Не зменшуючи загальності, нехай задано деякий безрозмірний растр, кожному пікселю якого поставлено у відповідність двійку індексів $\{(i, j)\}_{i,j \in \mathbf{Z}}$, що визначають його місцеположення. Позначимо $\{p_{i,j}\}_{i,j \in \mathbf{Z}}$ – для запису обчислювальних схем при роботі з послідовностями кольорових складових (червоною, зеленою та синьою) ЦЗ або, в більш загальному випадку, можемо говорити про обробку ЦЗ у градаціях сірого, тобто

$$p_{i,j} = 0,2125 \cdot pR_{i,j} + 0,7154 \cdot pG_{i,j} + 0,0721 \cdot pB_{i,j} \quad i, j \in \mathbf{Z},$$

де pR, pG, pB – червона, зелена та синя кольорові складові растра.

Якщо обирати в моделі (1) низькочастотний фільтр у вигляді згортки з *B*-сплайнами, то можливим є отримання аналогічних за властивостями операторів, що й на основі гауссіанів (4)–(7), проте вони будуть мати важливу перевагу – нижчу обчислювальну складність за рахунок того, що зазначені сплайн-оператори – лінійні. Отже, виходячи з моделі цифрових зображень на основі лінійних

комбінацій *B*-сплайнів різних порядків, близьких до інтерполяційних у середньому [6; 11],

$$S_{r,0}\left(p,t,q\right)=\sum_{i\in Z}\sum_{j\in Z}p_{i,j}B_{r,h_t}\left(t-ih_t\right)B_{r,h_q}\left(q-jh_q\right),\qquad(8)$$

де $B_{r,h}(\square)$ – *B*-сплайни *r*-го порядку за рівномірним розбиттям [12], можна очікувати більшої швидкодії алгоритмів розпізнавання на основі пошуку особливих точок зображення, інваріантних до обертань та зміни масштабу.

Тож, поставимо за мету запропонувати в даній роботі новий метод розпізнавання об'єктів на ЦЗ, в основу якого покладено ідею методу SIFT та модель ЦЗ у вигляді лінійної комбінації *B*-сплайнів.

Виклад основного матеріалу. Будемо називати «еталонним зображенням» фрагмент ЦЗ, який підлягає пошуку на інших зображеннях, а «тестовим зображенням» – ЦЗ, яке може містити шуканий об'єкт. Суть пропонованого методу полягає в тому, що спочатку для еталонного зображення визначається набір особливих точок та їх дескрипторів, на основі котрих буде здійснюватися розпізнавання. Далі на тестовому зображенні також визначають набір особливих точок та порівнюють з точками еталона.

Пошук особливих точок. Для початку до еталонного зображення, застосуємо лінійний оператор $Delta\left(p^{i,j}\right)$ у вигляді дискретної згортки послідовності $\left\{p_{i,j}\right\}_{i,j\in\mathbf{Z}}$ з маскою ∇S:

$$d\_p_{i,j}=Delta\left(p^{i,j}\right)=\sum_{ii=i-3}^{i+3}\sum_{jj=j-3}^{j+3}\nabla S_{\,ii-i,jj-j}\,p_{ii,jj},\quad i,j\in\mathbf{Z},$$

де $\left\{d\_p_{i,j}\right\}_{i,j\in\mathbf{Z}}$ – новоутворена послідовність для пошуку особливих точок;

$$\nabla S=\frac{1}{21233664}\begin{pmatrix}0,01 & 7,22 & 105,43 & 235,48 & \dots\\ 7,22 & 3738,28 & 37781,9 & 72695,6 & \dots\\ 105,43 & 37781,9 & 114745,93 & -47679,32 & \dots\\ 235,48 & 72695,6 & -47679,32 & -878100,32 & \dots\\ \vdots & \vdots & \vdots & \vdots & \ddots\end{pmatrix};$$

∇S – симетрична квадратна матриця розмірності (7×7), елементи якої подано з урахуванням симетрії.

Далі формуємо тривимірний масив

$\left\{\left(ext_l, i\_pos_l, j\_pos_l\right); l=\overline{1,M}\right\}$ обсягу M, що містить локальні мінімуми та максимуми $\left\{d\_p_{i,j}\right\}_{i,j\in\mathbf{Z}}$: якщо

$$d\_p_{i,j} = \max\left\{d\_p_{ii,jj}; ii=\overline{i-1,i+1}, jj=\overline{j-1,j+1}\right\}$$

або

$$d\_p_{i,j} = \min\left\{d\_p_{ii,jj}; ii=\overline{i-1,i+1}, jj=\overline{j-1,j+1}\right\},$$

то

$$i\_pos_l = i, \qquad j\_pos_l = j.$$

З використанням отриманих місцеположень локальних екстремумів обчислюємо значення деякого детектора особливої точки, на зразок (4)–(7), проте з урахуванням, що складові 2-джета (3)

$$L_t,\ L_q,\ L_{tt},\ L_{tq},\ L_{qq}$$

– лінійні оператори дискретних згорток масок часткових похідних моделі (8) з послідовністю $\left\{d\_p_{i,j}\right\}_{i,j\in\mathbf{Z}}$, розглянутих у роботі [13], наприклад:

$$L_t = \sum_{ii=i-1}^{i+1}\sum_{j=i-1}^{j+1} \gamma'_{x,ii-i,jj-j}\cdot d\_p_{ii,jj},$$

де

$$\gamma'_t = \frac{1}{16}\begin{pmatrix} -1 & -6 & -1 \\ 0 & 0 & 0 \\ 1 & 6 & 1 \end{pmatrix}.$$

Не зменшуючи загальності, розглянемо обрахування детектора на основі значення кривої рівня масштабування $\tilde{k}$ (7). Тобто для кожного локального екстремуму, положення якого визначено

$$i\_pos_l \text{ та } j\_pos_l,\ l=\overline{1,N},$$

можна отримати детектор особливої точки так:

$$ext_l = \tilde{k},\ l=\overline{1,M}.$$

Для остаточного відбору особливих точок, що придатні для розпізнавання, необхідно проаналізувати розподіл ймовірностей масиву значень $\left\{ext_l; l=\overline{1,M}\right\}$ та залишити ті, що задовольняють

умовам:

$$ext_l \le ext_{\alpha_1},\ l=\overline{1,M}, \qquad ext_{\alpha_1}=F^{-1}(\alpha_1)$$

та

$$ext_l \ge ext_{1-\alpha_2},\ l=\overline{1,N}, \qquad ext_{1-\alpha_2}=F^{-1}(1-\alpha_2),$$

де $F^{-1}(\bullet)$ – зворотна функція розподілу ймовірностей кривої рівня масштабування $\tilde{k}$ для конкретного контрольного об'єкта за $\left\{ext_l; l=\overline{1,M}\right\}$;

ext_{α_1}, $ext_{1-\alpha_2}$ – квантилі такого розподілу на його хвостах при деяких відносно малих ймовірностях α_1 та α_2, які для визначеності можна покласти

$$\alpha_1=0,01 \text{ та } \alpha_2=0,01.$$

Зауважимо, що при виборі для визначення особливостей іншого детектора, ніж крива рівня масштабу $\tilde{k}$, значення ймовірностей α_1, α_2 можуть бути іншими, залежно від форми функції щільності розподілу значень відповідного детектора. Окрім того, на вибір величини α_1 та α_2 може впливати розмір контрольного об'єкта та конкретний вигляд ЦЗ, тож для більш детальних рекомендацій з цього приводу потрібні окремі додаткові дослідження моделей розподілу значень детекторів особливих точок.

Визначення дескриптора особливої точки. Після того як визначено місцеположення особливих точок на еталонному зображенні у вигляді масиву $\left\{(i\_pos_l, j\_pos_l); l=\overline{1,N}\right\}$, де N – їх кількість, для кожної такої точки необхідно побудувати дескриптор, за допомогою якого буде відбуватися процедура розпізнавання. Побудова дескриптора особливої точки полягає у визначенні двох складових: магнітуди та орієнтації градієнта безпосередньо в точці та деякому її околі.

Усі обрахунки відбуваються зі згладженими значеннями еталонного об'єкта. Тобто на основі $\left\{p_{i,j}\right\}_{i,j\in\mathbf{Z}}$ та масок низькочастотних фільтрів [14] на основі моделі (8) отримуємо згладжене зображення $\left\{pL_{i,j}\right\}_{i,j\in\mathbf{Z}}$, наприклад,

$$pL_{i,j} = \sum_{ii=i-3}^{i+3} \sum_{jj=j-3}^{j+3} \gamma_{6\ ii-i,jj-j}\, p_{ii,jj}\ ,\quad i,j \in \mathbf{Z}\ ,$$

де

$$\gamma_6 = \frac{1}{21233664} \begin{pmatrix} 0,01 & 7,22 & 105,43 & 235,48 & \dots \\ 7,22 & 5212,84 & 76120,46 & 170016,56 & \dots \\ 105,43 & 76120,46 & 1111548,49 & 2482665,64 & \dots \\ 235,48 & 170016,56 & 2482665,64 & 5545083,04 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Далі, з урахуванням індексів місця розташування особливих точок (i_pos_l, j_pos_l), $l = \overline{1,N}$, визначаємо для усіх

$$pL_{i,j}\ ,\ i = \overline{i\_pos-3, i\_pos+3}\ ,\ j = \overline{j\_pos-3, j\_pos+3}$$

магнітуду градієнта

$$m(i-i\_pos, j-j\_pos) = \sqrt{L_t^2 + L_q^2}$$

та орієнтацію у вигляді кута вектора краю в особливій точці

$$\theta(i-i\_pos, j-j\_pos) = arctg\left(\frac{L_q}{L_t}\right),$$

де L_t, L_q – лінійні оператори дискретних згорток масок часткових похідних моделі (8) з послідовністю $\left\{Lp_{i,j}\right\}_{i,j\in\mathbf{Z}}$, розглянутих у роботі [13], наприклад :

$$L_q = \sum_{ii=i-1}^{i+1} \sum_{j=i-1}^{j+1} \gamma'_{y,ii-i,jj-j} \cdot Lp_{ii,jj}\ ,$$

де

$$\gamma'_q = \frac{1}{16} \begin{pmatrix} -1 & 0 & 1 \\ -6 & 0 & 6 \\ -1 & 0 & 1 \end{pmatrix}.$$

Отже, для кожної особливої точки, що розташована за місцеположенням індексів (i_pos_l, j_pos_l), $l = \overline{1,N}$ маємо дескриптор, який містить по 49 значень магнітуди та орієнтації градієнта в самій точці та точках її околу:

$$m\left(i-i\_pos, j-j\_pos\right), \qquad \theta\left(i-i\_pos, j-j\_pos\right),$$

$$i=\overline{i\_pos-3, i\_pos+3}, \qquad j=\overline{j\_pos-3, j\_pos+3}.$$

Наостанок, отримані значення дескрипторів зважуємо деякою ваговою функцією, щоб врахувати віддаленість точок околу від особливої. В якості такої вагової функції можна взяти, наприклад, наведену вище маску γ_6, тобто фактичні значення дескриптора будуть такі:

$$m\left(i-i\_pos, j-j\_pos\right)=m\left(i-i\_pos, j-j\_pos\right)\gamma_{6\ i-i\_pos, j-j\_pos},$$

$$\theta\left(i-i\_pos, j-j\_pos\right)=\theta\left(i-i\_pos, j-j\_pos\right)\gamma_{6\ i-i\_pos, j-j\_pos},$$

$$i=\overline{i\_pos-3, i\_pos+3}, \qquad j=\overline{j\_pos-3, j\_pos+3}.$$

У подальшому для розпізнавання будемо використовувати лише багатовимірний масив дескрипторів особливих точок еталонного зображення

$$\left\{\left(m_l(i,j), \theta_l(i,j); i=\overline{-3,3}, j=\overline{-3,3}\right); l=\overline{1,N}\right\}. \tag{9}$$

Після детектування особливих точок за допомогою розглянутих операторів постає задача пошуку особливих точок, які є спільними для еталонного та тестового зображень.

Нехай

$A=\left\{a_l, l=\overline{1,N}\right\}$ – набір особливих точок еталонного зображення,

$B=\left\{b_k, k=\overline{1,M}\right\}$ – набір особливих точок тестового зображення,

$\left\{\left(i\_pos_k, j\_pos_k\right), k=\overline{1,M}\right\}$ – масив індексів місця розташування особливих точок на тестовому зображенні,

$$N \le M,$$

де кожне a_l та b_k визначаються на зразок елементів масиву (9) векторами двійок дескрипторів, а саме – магнітуди градієнта та його орієнтації.

У якості міри відстані між дескрипторами будемо використовувати метрику на основі евклідової відстані між складовими векторів детекторів:

$$d_{lk}=\sqrt{d_{lk}^{m}+d_{lk}^{\theta}},$$

де

$$d_{lk}^{m} = \sum_{i=-3}^{3}\sum_{j=-3}^{3}\left(m^{A(l)}(i,j) - m^{B(k)}(i,j)\right)^2;$$

$$d_{lk}^{\theta} = \sum_{i=-3}^{3}\sum_{j=-3}^{3}\left(\theta^{A(l)}(i,j) - \theta^{B(k)}(i,j)\right)^2;$$

$$l = \overline{1,N}; \qquad k = \overline{1,M};$$

$m^{A(l)}(\cdot,\cdot)$, $m^{B(k)}(\cdot,\cdot)$ – магнітуди l-ї точки еталонного та k-ї точки тестового зображень;

$\theta^{A(l)}(\cdot,\cdot)$, $\theta^{B(k)}(\cdot,\cdot)$ – орієнтація градієнтів l-ї точки еталонного та k-ї точки тестового зображень.

Далі, для кожної l-ї точки еталонного зображення найближчу k-ту точку тестового зображення знаходимо з умови:

$$d_l^{\min} = \min_k \{d_{lk}\}, \tag{10}$$

тим самим визначаючи двовимірний масив місця розташування індексів особливих точок тестового зображення, що відповідають умові (10):

$$\left\{(i\_pos_k, j\_pos_k), k = \overline{1,N}\right\}, \; N \le M. \tag{11}$$

Завжди існує ймовірність, що отримані точки (11) можуть як належати зображенню шуканого об'єкта на тестовому зображенні, так і бути просто випадково схожими на такі особливі точки. Якщо оцінити двовимірну щільність розподілу точок (11) на тестовому зображенні, то їх компактне розташування і буде визначати положення шуканого об'єкта. Іншими словами, мова йде про пошук областей, де така оцінка щільності буде мати локальні максимуми або один глобальний максимум, якщо об'єкт пошуку лише один на тестовому зображенні. В якості оцінки двовимірної щільності розподілу за масивом (11) можна обрати сплайн-оцінку на основі сплайнів типу (8) [11] або обмежитися побудовою та аналізом гістограми відносних частот для локалізації прямокутних областей ймовірного розташування шуканого об'єкта.

Для реалізації останнього підходу пропонується таке. Нехай задано деякі (довільні)

$$h_t > 0, \quad h_q > 0,$$

причому можна обирати ці величини близькими до лінійних розмірів (у пікселях) шуканого об'єкта, але не обов'язково. Як завжди, будемо вважати, що відлік індексів пікселів на цифровому зображенні починається з лівого верхнього кута. Задамо розбиття Δ_{h_t,h_q} тестового

зображення на прямокутні області, верхні ліві кути котрих визначаються точками з індексами

$$\Delta_{h_t,h_q} : \left(ii \cdot h_t, jj \cdot h_q\right), \qquad ii = \overline{0, Nn-1}, \; jj = \overline{0, Mm-1},$$

де Nn, Mm – кількість областей за відповідними напрямками.

Відносні частоти розподілу, які визначають емпіричну ймовірність появи особливої точки з масиву (11) в конкретній з областей розбиття Δ_{h_t,h_q}, отримуємо так:

$$f_{ii,jj} = \frac{1}{N}\sum_{k=1}^{N} I_k, \quad ii = \overline{0, Nn-1}, \; jj = \overline{0, Mm-1},$$

де

$$I_k = \begin{cases} 1, & якщо \quad ii = \left[\dfrac{i\_pos_k}{h_t}\right] \quad та \quad jj = \left[\dfrac{j\_pos_k}{h_q}\right], \\ 0, & у \quad іншому \quad випадку; \end{cases}$$

$[\square]$ – ціла частина.

Деяка (ii, jj) та область розбиття Δ_{h_t,h_q}, де виконується умова

$$\max_{ii,jj}\left\{f_{ii,jj}\right\}$$

і буде містити геометричне розташування шуканого об'єкта.

Якщо припускати, що об'єкт пошуку на тестовому зображенні може розташовуватися в межах декількох сусідніх областей, то для їх локалізації достатньо відібрати області, де виконується умова

$$f_{ii,jj} \geq f_{порогове}, \tag{12}$$

де $f_{порогове}$ – деяке порогове значення.

Величини $f_{ii,jj}$, $ii = \overline{0, Nn-1}$, $jj = \overline{0, Mm-1}$ надають емпіричну ймовірність появи об'єкта або його частини пошуку в конкретній області розбиття Δ_{h_t,h_q}, тож, якщо така область для об'єкта лише одна, то ймовірність $\mathrm{P}_{рез}$ того, що знайдена область містить об'єкт

$$\mathrm{P}_{рез} = \max_{ii,jj}\left\{f_{ii,jj}\right\},$$

а якщо об'єкт займає декілька областей, то

$$\mathrm{P}_{рез} = \sum_{ii=0}^{Nn-1}\sum_{jj=0}^{Mm-1} f_{ii,jj} \cdot J_{ii,jj},$$

де

$$J_{ii,jj} = \begin{cases} 1, & f_{ii,jj} \geq f_{порогове}, \\ 0, & f_{ii,jj} < f_{порогове}. \end{cases}$$

Ймовірність P_α похибки локалізації об'єкта пошуку не складно оцінити за виразом

$$P_\alpha = 1 - P_{рез}. \qquad (13)$$

Введення ймовірності (13) дозволяє сформулювати статистичну гіпотезу про виявлення шуканого об'єкта на тестовому зображенні та її альтернативну гіпотезу – про відсутність такого об'єкта. Задаючись рівнем значущості α (ймовірність похибки першого роду), отримуємо умову відхилення головної гіпотези:

$$P_\alpha > \alpha .$$

Зауважимо, що викладений підхід до локалізації місцеположення шуканого об'єкта нескладно узагальнити і на випадок пошуку більшої кількості його візуалізацій на тестовому зображенні.

Висновок. На основі відомої ідеї розпізнавання об'єктів на ЦЗ з використанням особливих точок (метод SIFT) у даній роботі запропоновано новий метод, що базується на сплайн-моделі зображення у вигляді лінійної комбінації *B*-сплайнів, близьких до інтерполяційних у середньому. На відміну від описаних раніше модифікацій методу SIFT, запропонований має суттєво меншу обчислювальну складність та дозволяє оцінювати ймовірність помилкової ідентифікації об'єкта пошуку.

Подальші дослідження можуть полягати у вивченні законів розподілу реалізацій диференціальних інваріантів (4)–(7) з метою вдосконалення критеріїв відбору особливих точок та отриманні нових часткових похідних моделі (8) на основі *B*-сплайнів порядку вище 2-го, що також має сприяти більш адекватному розпізнаванню об'єктів пошуку.

Бібліографічні посилання

1. Turk M., Pentland A. Eigenfaces for Recognition. // Journal of cognitive neuroscience. Vol. 3, N. 1. Massachusetts Institute of technology, 1996. URL: http://face-rec.org/algorithms/PCA/jcn.pdf.

2. Etemad K., Chellappa R. Discriminant Analysis for Recognition of Human Face Imager // Journal of the Optical Society of America A. Vol. 14. No 8. 1997. P. 1724–1733.

3. Active shape models – their training and application / T. Cootes et al. // Computer Vision and Image Understanding. Jan.1995. 61 (1). P. 38–59.

4. Lowe D. G. Object recognition from local scale-invariant features. // Proceedings of the International Conference on Computer Vision. Corfu. Greece. 1999. P. 1150–1157.

5. Bay H., Tuytelaars T., van Gool L. SURF: Speeded up robust features // Proc. of the 9th European Conference on Computer Vision (ECCV'06). Springer Lecture Notes on Computer Science. 2006. V. 3951. Part 1. P. 404–417.

6. Приставка П. О., Рябий М. О. Модель реалістичних зображень на основі двовимірних сплайнів, близьких до інтерполяційних у середньому // Наукоємні технології. 2012. № 3 (15). С. 67–71.

7. Кухаренко Б. Г. Алгоритмы анализа изображений для определения локальных особенностей и распознавания объектов и панорам // Информационные технологии. № 7. 2011. Приложение. 32 с.

8. Kalal Z., Mikolaiczyk K., Matas J. Tracking-Learning-Detection // IEEE Transaction Pattern Analysis&Machine Intelligence. Vol. 34. N. 7. 2012. P.1409–1422.

9. Tuytelaars T. Local invariant feature detectors. A survey. // Foundations and Trends in Computer Graphics and Vision. Vol. 3. N. 3. 2007. P. 177–280.

10. Beaudet P. R. Rotationally invariant image operators // Proc. of the International Joint Conference on Pattern Recognition. 1978. P. 579–583.

11. Приставка П. О. Поліноміальні сплайни при обробці даних. Дніпропетровськ. 2004. 236 с.

12. Лигун А. А., Шумейко А. А. Асимптотические методы восстановления кривых. Киев. 1997. 358 с.

13. Приставка П. О. Визначення особливостей зображень на основі комбінацій *B*-сплайнів другого порядку, близьких до інтерполяційних у середньому // Актуальні проблеми автоматизації та інформаційних технологій. Т. 19. 2015. С. 67–77.

14. Приставка П. О. Обчислювальні аспекти застосування поліноміальних сплайнів при побудові фільтрів // Актуальні проблеми автоматизації та інформаційних технологій. Т. 10. Дніпропетровськ. 2006. С. 3–14.

Надійшла до редколегії 07.09.16.

УДК 519.254:519.652(045)

П.О. Приставка, О.Г. Чолишкіна, Б.І. Мартюк

Національний авіаційний університет

ДОСЛІДЖЕННЯ ПОХІДНИХ ЛІНІЙНОЇ КОМБІНАЦІЇ *B*-СПЛАЙНІВ ЧЕТВЕРТОГО ПОРЯДКУ

Отримано похідні першого та другого порядку для поліноміальних сплайнів, близьких до інтерполяційних у середньому, на основі *B*-сплайнів четвертого порядку, визначених на рівномірному розбитті вісі аргументу. Досліджено властивості отриманих похідних, подано теореми про норми та якості апроксимації.

Ключові слова: *сплайн, похідні, аналогові сигнали.*

Получены производные первого и второго порядка для полиномиальных сплайнов, близких к интерполяционным в среднем, на основании *B*-сплайнов четвертого порядка, определенных на равномерном разбиении оси аргумента. Исследованы особенности полученных производных, представлены теоремы о нормах и качестве аппроксимации.

Ключевые слова: *сплайн, производные, аналоговые сигналы.*

Obtained derivatives of the first and second-order polynomial splines, interpolation close to the average, based on the *B*-splines of the fourth order, defined on a uniform partition argument axis. The features of the obtained derivatives represented theorem on approximation of standards and quality.

Keywords: *spline, derivative, analog signals*.

Постановка проблеми. При фіксації аналогового сигналу має місце таке. Нехай $\phi(t)$ – функція імпульсного відклику системи, що реєструє сигнал $p(t)$. Враховуючи технічні властивості систем реєстрації, результатом згортки сигналу та функції відклику буде значення, усереднене на інтервалі дискретизації:

$$(p*\phi)(ih)=\frac{1}{h}\int_{ih-\frac{h}{2}}^{ih+\frac{h}{2}} p(t)\phi(t-ih)\,dt=\overline{p}_i\,.$$

Тоді цифровий сигнал має такий вигляд:

$$p_i = \overline{p}_i + \varepsilon_i,\ i \in \mathbf{Z}, \qquad (1)$$

де ε_i – випадкова вада.

У задачах обробки цифрових сигналів, що задані співвідношенням (1), в якості моделі $p(t)$ є потреба використовувати наближення із властивостями імпульсного нерекурсивного низькочастотного фільтра, наприклад, лінійні комбінації *B*-сплайнів, близькі до інтерполяційних у середньому [1].

Відомо [2; 5], що будь-який *B*-сплайн порядку вище першого може бути використаний для визначення коротковіконного перетворення Фур'є. Актуальним є дослідження лінійних комбінацій *B*-сплайнів вищого порядку та їх похідних, що дозволить очікувати більш високі згладжувальні властивості при відносно низькій обчислювальній складності.

Для пошуку особливостей аналогового сигналу можна використовувати першу та другу похідні лінійних комбінацій *B*-сплайнів, відповідно, постає питання дослідження цих похідних. Так, у роботах [3; 4] досліджено властивості похідних локальних поліноміальних сплайнів на основі *B*-сплайнів другого та третього порядків. Тому актуальною є потреба отримання та дослідження похідних лінійних комбінацій *B*-сплайнів більш високого порядку, зокрема четвертого, що обумовлено властивостями відповідних *B*-сплайнів [1].

Аналіз досліджень та постановка задачі. Задачу відтворення гладких функцій на основі лінійних комбінацій *B*-сплайнів висвітлено у роботах І. Шоенберга, К. де Бора, М.П. Корнійчука та ін. Увагу поліноміальним сплайнам, визначеним на локальних носіях, близьким до інтерполяційних у середньому, приділено А. О. Лигуном, О. О. Шумейко, В. В. Кармазіною [2–4] та в авторських роботах [5].

Нехай з кроком $h > 0$ задано розбиття дійсної вісі $\Delta_h : t_i = ih$, $i \in Z$, у кожній точці якого отримано значення деякої неперервної функції $p(t) \in C^r$, $r \geq 2$, визначеної на $\mathbf{R}_1(-\infty;\infty)$. Вважають, що інформацію про функцію $p(t)$, яка підлягає відтворенню, задано у вузлах розбиття Δ_h у вигляді інтеграла $\overline{p}_i = \frac{1}{h}\int\limits_{(i-0,5)h}^{(i+0,5)h} p(t)\,dt$, при цьому істинне значення функції $p(t)$ у вузлах визначається аналогічно виразу (1).

Для апроксимації функції $p(t)$ за значеннями типу (1) у вузлах розбиття Δ_h вводяться такі поліноміальні сплайни на основі *B*-сплайнів, що є близькими до інтерполяційних у середньому [5]:

$$S_{4,0}(p,t)=\sum_{i\in Z} p_i B_{4,h}\left(t-(i+0,5)h\right),$$

$$S_{4,1}(p,t)=\sum_{i\in Z}\left(p_i-\frac{1}{4}\Delta^2 p_i\right)B_{4,h}\left(t-(i+0,5)h\right),$$

$$S_{4,2}(p,t)=\sum_{i\in Z}\left(p_i-\frac{1}{4}\Delta^2 p_i+\frac{13}{240}\Delta^4 p_i\right)B_{4,h}\left(t-(i+0,5)h\right),$$

де *B*-сплайн $B_{r,h}(t)$ порядку r ($r\ge 1$) визначається рекурентно таким чином: якщо

$$B_{0,h}(t)=\begin{cases}0, & t\notin[-h/2;h/2],\\ 1, & t\in[-h/2;h/2],\end{cases}$$

то

$$B_{r,h}(t)=\frac{1}{h}\int_{t-h/2}^{t+h/2} B_{r-1,h}(\tau)\,d\tau. \qquad (2)$$

Таким чином, *B*-сплайн четвертого порядку має вигляд [5]:

$$B_{4,h}(t)=\begin{cases}0, & t\notin[-5h/2;5h/2],\\ (t/h+5/2)^4/24, & t\in[-5h/2;-3h/2],\\ \psi(t/h)-5(t/h+1/2)^4/12, & t\in[-3h/2;-h/2],\\ \psi(t/h), & t\in[-h/2;h/2],\\ \psi(t/h)-5(t/h-1/2)^4/12, & t\in[h/2;3h/2],\\ (t/h-5/2)^4/24, & t\in[3h/2;5h/2].\end{cases} \qquad (3)$$

де $\psi(t)=\frac{115}{192}-\frac{5}{8}t^2+\frac{1}{4}t^4$.

Для апроксимації функції $p(t)$ за значеннями типу (1) у вузлах розбиття Δ_h зручно використовувати лінійні комбінації *B*-сплайнів (3):

$$S_{4,0}(p,t)=\frac{1}{384}\left(p_{i-2}-4p_{i-1}+6p_i-4p_{i+1}+p_{i+2}\right)x^4+\frac{1}{96}(-p_{i-2}+$$

$$+2p_{i-1}-2p_{i+1}+p_{i+2})x^3+\frac{1}{64}\left(p_{i-2}+4p_{i-1}-10p_i+4p_{i+1}+p_{i+2}\right)x^2+$$

$$+\frac{1}{96}\left(-p_{i-2}-22p_{i-1}+22p_{i+1}+p_{i+2}\right)x+\frac{1}{384}(p_{i-2}+76p_{i-1}+$$

$$+230p_i+76p_{i+1}+p_{i+2}), \qquad (4)$$

$$S_{4,1}(p,t)=\frac{1}{1536}\left(-p_{i-3}+10p_{i-2}-31p_{i-1}+44p_i-31p_{i+1}+10p_{i+2}-\right.$$

$$\left.-p_{i+3}\right)x^4+\frac{1}{384}\left(p_{i-3}-8p_{i-2}+13p_{i-1}-13p_{i+1}+8p_{i+2}-p_{i+3}\right)x^3+$$

$$+\frac{1}{256}\left(-p_{i-3}+2p_{i-2}+33p_{i-1}-68p_i+33p_{i+1}+2p_{i+2}-p_{i+3}\right)x^2+$$

$$+\frac{1}{384}\left(p_{i-3}+16p_{i-2}-131p_{i-1}+131p_{i+1}-16p_{i+2}-p_{i+3}\right)x+$$

$$+\frac{1}{1536}\left(-p_{i-3}-70p_{i-2}+225p_{i-1}+1228p_i+225p_{i+1}-70p_{i+2}-p_{i+3}\right), \qquad (5)$$

$$S_{4,2}(p,t)=\frac{1}{92160}(13p_{i-4}-164p_{i-3}+964p_{i-2}-2588p_{i-1}+3550p_i-$$

$$-2588p_{i+1}+964p_{i+2}-164p_{i+3}+13p_{i+4})x^4+\frac{1}{23040}(-13p_{i-4}+$$

$$+138p_{i-3}-662p_{i-2}+962p_{i-1}-962p_{i+1}+662p_{i+2}-138p_{i+3}+13p_{i+4})x^3+$$

$$+\frac{1}{46080}(39p_{i-4}-180p_{i-3}-420p_{i-2}+8436p_{i-1}-15750p_i+8436p_{i+1}-$$

$$-420p_{i+2}-180p_{i+3}+39p_{i+4})x^2+\frac{1}{23040}(-13p_{i-4}-174p_{i-3}+$$

$$+2026p_{i-2}-9238p_{i-1}+9238p_{i+1}-2026p_{i+2}+174p_{i+3}+13p_{i+4})x+$$

$$+\frac{1}{92160}(13p_{i-4}+876p_{i-3}-5084p_{i-2}+8404p_{i-1}+83742p_i+$$

$$+8404p_{i+1}-5084p_{i+2}-876p_{i+3}+13p_{i+4}). \qquad (6)$$

Нехай у вузлах розбиття Δ_h для значень деякої гладкої неперервної функції $p(t)$ виконується умова (1). Тоді для сплайну (4–6) має місце така рівність:

$$S_{4,u}(p,t)=S_{4,u}(\bar{p},t)+S_{4,u}(\varepsilon,t),\ u=0,1,2\ .$$

Оцінка якості відтворення функції $p(t)$ зводиться до оцінки відхилення

$$\left|p(t)-S_{4,u}(p,t)\right|=\left|p(t)-S_{4,u}(\overline{p},t)-S_{4,u}(\varepsilon,t)\right|,$$

або для $|\varepsilon_i|<\varepsilon$, $i\in Z$, до оцінки нерівності

$$\left|p(t)-S_{4,u}(p,t)\right|\le\left|p(t)-S_{4,u}(\overline{p},t)\right|+\varepsilon\left\|S_{4,u}\right\|, \tag{7}$$

де

$$\left\|S_{4,u}\right\|=\sup_{|\varepsilon_i|}\max_{t}\left|S_{4,u}(\varepsilon,t)\right|, \tag{8}$$

норма сплайн-оператора $S_{4,u}(p,t)$.

Поставимо за мету даної роботи отримати явний вигляд похідних сплайнів $S_{4,0}(p,t)$, $S_{4,1}(p,t)$, $S_{4,2}(p,t)$ та знаходження норм і якості апроксимації для даних похідних.

Виклад основного матеріалу. Отримаємо явний вигляд 1-ї та 2-ї похідної для сплайнів $S_{4,0}(p,t)$, $S_{4,1}(p,t)$, $S_{4,2}(p,t)$.

$$S'_{4,0}(p,t)=\frac{1}{96}(p_{i-2}-4p_{i-1}+6p_i-4p_{i+1}+p_{i+2})x^3+\frac{1}{32}(-p_{i-2}+$$

$$+2p_{i-1}-2p_{i+1}+p_{i+2})x^2+\frac{1}{32}(p_{i-2}+4p_{i-1}-10p_i+4p_{i+1}+p_{i+2})x+$$

$$+\frac{1}{96}(-p_{i-2}-22p_{i-1}+22p_{i+1}+p_{i+2}), \tag{9}$$

$$S''_{4,0}(p,t)=\frac{1}{32}(p_{i-2}-4p_{i-1}+6p_i-4p_{i+1}+p_{i+2})x^2+\frac{1}{16}(-p_{i-2}+$$

$$+2p_{i-1}-2p_{i+1}+p_{i+2})x+\frac{1}{32}(p_{i-2}+4p_{i-1}-10p_i+4p_{i+1}+p_{i+2}), \tag{10}$$

$$S'_{4,1}(p,t)=\frac{1}{384}(-p_{i-3}+10p_{i-2}-31p_{i-1}+44p_i-31p_{i+1}+10p_{i+2}-$$

$$-p_{i+3})x^3+\frac{1}{128}(p_{i-3}-8p_{i-2}+13p_{i-1}-13p_{i+1}+8p_{i+2}-p_{i+3})x^2+$$

$$+\frac{1}{128}(-p_{i-3}+2p_{i-2}+33p_{i-1}-68p_i+33p_{i+1}+2p_{i+2}-p_{i+3})x+$$

$$+\frac{1}{384}(p_{i-3}+16p_{i-2}-131p_{i-1}+131p_{i+1}-16p_{i+2}-p_{i+3}), \tag{11}$$

$$S''_{4,1}(p,t)=\frac{1}{128}(-p_{i-3}+10p_{i-2}-31p_{i-1}+44p_i-31p_{i+1}+10p_{i+2}-$$
$$-p_{i+3})x^2+\frac{1}{64}(p_{i-3}-8p_{i-2}+13p_{i-1}-13p_{i+1}+8p_{i+2}-p_{i+3})x+$$
$$+\frac{1}{128}(-p_{i-3}+2p_{i-2}+33p_{i-1}-68p_i+33p_{i+1}+2p_{i+2}-p_{i+3}), \qquad (12)$$

$$S'_{4,2}(p,t)=\frac{1}{23040}(13p_{i-4}-164p_{i-3}+964p_{i-2}-2588p_{i-1}+3550p_i-$$
$$-2588p_{i+1}+964p_{i+2}-164p_{i+3}+13p_{i+4})x^3+\frac{1}{7680}(-13p_{i-4}+138p_{i-3}-$$
$$-662p_{i-2}+962p_{i-1}-962p_{i+1}+662p_{i+2}-138p_{i+3}+13p_{i+4})x^2+$$
$$+\frac{1}{23040}(39p_{i-4}-180p_{i-3}-420p_{i-2}+8436p_{i-1}-15750p_i+8436p_{i+1}-$$
$$-420p_{i+2}-180p_{i+3}+39p_{i+4})x+\frac{1}{23040}(-13p_{i-4}-174p_{i-3}+2026p_{i-2}-$$
$$-9238p_{i-1}+9238p_{i+1}-2026p_{i+2}+174p_{i+3}+13p_{i+4}), \qquad (13)$$

$$S''_{4,2}(p,t)=\frac{1}{7680}(13p_{i-4}-164p_{i-3}+964p_{i-2}-2588p_{i-1}+3550p_i-$$
$$-2588p_{i+1}+964p_{i+2}-164p_{i+3}+13p_{i+4})x^2+\frac{1}{3840}(-13p_{i-4}+138p_{i-3}-$$
$$-662p_{i-2}+962p_{i-1}-962p_{i+1}+662p_{i+2}-138p_{i+3}+13p_{i+4})x+$$
$$+\frac{1}{23040}(39p_{i-4}-180p_{i-3}-420p_{i-2}+8436p_{i-1}-15750p_i+8436p_{i+1}-$$
$$-420p_{i+2}-180p_{i+3}+39p_{i+4}). \qquad (14)$$

Подальша задача оцінки якості відтворення $p(t)$ складається з двох етапів: знаходження норм похідних сплайн-операторів $S_{4,0}(p,t)$, $S_{4,1}(p,t)$, $S_{4,2}(p,t)$ і задача визначення якості апроксимації.

Проведемо оцінку норм перших та других похідних сплайнів $S_{4,0}(p,t)$, $S_{4,1}(p,t)$, $S_{4,2}(p,t)$.

Теорема 1. Для сплайну $S'_{4,0}(p,t)$ є вірним:

$$\left\|S'_{4,0}(p,t)\right\|_C=\frac{2}{3}\left\|p(t)\right\|_C.$$

Доведення. За визначенням норми оператора лінійного нормового простору для похідної сплайну $S'_{4,0}(p,t)$ маємо:

$$\left\| S'_{4,0}(p,t) \right\|_C = \sup_{p_i} \max_t \left| S'_{4,0}(p_i,t) \right|.$$

Згрупуємо сплайн (9) відносно p_i:

$$S'_{4,0}(p,t) = \frac{1}{96}\left(-(1-x)^3 p_{i-2} + \left(-22+12x+6x^2-4x^3\right) p_{i-1} + \right.$$
$$\left. + \left(-30x+6x^3\right) p_i + \left(22+12x-6x^2-4x^3\right) p_{i+1} + (1+x)^3 p_{i+2}\right). \quad (15)$$

З представлення сплайну $S'_{4,0}(p,t)$ у вигляді (15) слідує, що

$$\left\| S'_{4,0}(p,t) \right\|_C \le \frac{1}{96} \left\| p(t) \right\|_C \max_{|x|\le 1} A(x),$$

де

$$A(x) = |1-x|^3 + \left|-22+12x+6x^2-4x^3\right| + \left|-30x+6x^3\right| +$$
$$+ \left|22+12x-6x^2-4x^3\right| + |1+x|^3.$$

Враховуючи, що функція $A(x)$ парна, для знаходження її максимуму достатньо розглянути її для $x \in [0;1]$. Для цих x, зважаючи, що вирази під знаком модуля більші нуля, маємо:

$$\max_{x \in [0;1]} A(x) = A(1) = 64.$$

Таким чином,

$$\left\| S'_{4,0}(p,t) \right\|_C \le \frac{2}{3} \left\| p(t) \right\|_C, \quad (16)$$

з іншого боку, при $x = 0$ з (9) маємо

$$S'_{4,0}(p,t) = \frac{1}{96}(-p_{i-2} - 22p_{i-1} + 22p_{i+1} + p_{i-2}).$$

Тоді для деякого часткового випадку

$$p^*_{i-2} = p^*_{i-1} = p^*_i = p^*_{i+1} = p^*_{i+2} = p^*,$$

то

$$\left\| S'_{4,0}(p,t) \right\|_C \ge \left\| S'_{4,0}\left(p^*,t\right) \right\|_C \ge \left| S'_{4,0}\left(p^*,t\right) \right| = \frac{2}{3} \left\| p(t) \right\|_C.$$

Разом з нерівністю (16) це співвідношення доводить сформовану теорему.

Теорема 2. Якщо $p(t) \in C^5$, то при $h \to 0$ рівномірно по t має місце асимптотична рівність

$$p'(t)-S'_{4,0}(\overline{p},t)=-\frac{1}{4}p'''(t)h^2-\frac{1}{4}p^{(4)}(t)h^2\tau-$$
$$-p^{(5)}(t)\left(\frac{161}{5760}h^4-\frac{7}{48}h^2\tau^2+\frac{1}{24}\tau^4\right)+\mathrm{o}\left(h^3\right),$$

де

$$\tau=t-(i-0,5)h.$$

Доведення: Розкладемо сплайн $S'_{4,0}(\overline{p},t)$ в ряд Тейлора в околі точки $(i-0,5)h$ з урахуванням $S'_{4,0}(\overline{p},t)\in C^3$.

$$S'_{4,0}(\overline{p},t)=S'_{4,0}\left(\overline{p},t^*\right)+S''_{4,0}\left(\overline{p},t^*\right)\tau+S'''_{4,0}\left(\overline{p},t^*\right)\frac{\tau^2}{2!}+$$
$$+S^{(4)}_{4,0}\left(\overline{p},t^*\right)\frac{\tau^3}{3!}+o(h^3), \qquad (17)$$

де

$$S'_{4,0}\left(\overline{p},t^*\right)=\frac{1}{48h}(-\overline{p}_{i-2}-22\overline{p}_{i-1}+22\overline{p}_{i+1}-\overline{p}_{i+2}), \qquad (18)$$

$$S''_{4,0}\left(\overline{p},t^*\right)=\frac{1}{8h^2}(\overline{p}_{i-2}+4\overline{p}_{i-1}-10\overline{p}_i+4\overline{p}_{i+1}+\overline{p}_{i+2}), \qquad (19)$$

$$S'''_{4,0}\left(\overline{p},t^*\right)=\frac{1}{2h^3}(-\overline{p}_{i-2}+2\overline{p}_{i-1}-2\overline{p}_{i+1}+\overline{p}_{i+2}), \qquad (20)$$

$$S^{(4)}_{4,0}\left(\overline{p},t^*\right)=\frac{1}{h^4}(\overline{p}_{i-2}-4\overline{p}_{i-1}+6\overline{p}_i-4\overline{p}_{i+1}+\overline{p}_{i+2}), \qquad (21)$$

$$t^*=(i-0,5)h.$$

З урахуванням розкладу $S'_{4,0}(\overline{p},t)$ та $p(t)\in C^5$ в ряд Тейлора в околі точки $(i-0,5)h$, нескладно отримати, що для $\forall p(t)\in C^5$ при $h\to 0$ рівномірно по t має місце асимптотична рівність

$$p'(t)=p'\left(t^*\right)+p''\left(t^*\right)\tau+p'''\left(t^*\right)\frac{\tau^2}{2!}+p^{(4)}\left(t^*\right)\frac{\tau^3}{3!}+$$
$$+p^{(5)}\left(t^*\right)\frac{\tau^4}{4!}+o(h^4), \qquad (22)$$

причому:

$$\overline{p}_{i\pm2} = p\left(t^*\right) \pm 2p'\left(t^*\right)h + \frac{49}{24}p''\left(t^*\right)h^2 \pm \frac{17}{12}p'''\left(t^*\right)h^3 +$$
$$+\frac{1441}{1920}p^{(4)}\left(t^*\right)h^4 \pm \frac{931}{2880}p^{(5)}\left(t^*\right)h^5 + o(h^5),$$
$$\overline{p}_{i\pm1} = p\left(t^*\right) \pm p'\left(t^*\right)h + \frac{13}{24}p''\left(t^*\right)h^2 \pm \frac{5}{24}p'''\left(t^*\right)h^3 +$$
$$+\frac{121}{1920}p^{(4)}\left(t^*\right)h^4 \pm \frac{91}{5760}p^{(5)}\left(t^*\right)h^5 + o(h^5),$$
$$\overline{p}_i = p\left(t^*\right) + \frac{1}{24}p''\left(t^*\right)h^2 + \frac{1}{1920}p^{(4)}\left(t^*\right)h^4 + o(h^5).$$

Підставляючи ці рівності в рівності (18) – (21), отримаємо:

$$S'_{4,0}(\overline{p},t^*) = p'\left(t^*\right) + \frac{1}{4}p'''\left(t^*\right)h^2 + \frac{161}{5760}p^{(5)}\left(t^*\right)h^4 + o(h^5), \quad (22)$$

$$S''_{4,0}(\overline{p},t^*) = p''\left(t^*\right) + \frac{1}{4}p^{(4)}\left(t^*\right)h^2 + o(h^5), \quad (23)$$

$$S'''_{4,0}(\overline{p},t^*) = p'''\left(t^*\right) + \frac{7}{24}p^{(5)}\left(t^*\right)h^2 + o(h^5), \quad (24)$$

$$S^{(4)}_{4,0}(\overline{p},t^*) = p^{(4)}\left(t^*\right) + o(h^5). \quad (25)$$

Тоді підставляючи рівності (22) – (25) в (17), будемо мати:

$$S'_{4,0}(\overline{p},t) = \left(p'\left(t^*\right) + \frac{1}{4}p'''\left(t^*\right)h^2 + \frac{161}{5760}p^{(5)}\left(t^*\right)h^4\right) +$$
$$+\left(p''\left(t^*\right) + \frac{1}{4}p^{(4)}\left(t^*\right)h^2\right)\tau + \left(p'''\left(t^*\right) + \frac{7}{24}p^{(5)}\left(t^*\right)h^2\right)\frac{\tau^2}{2!} +$$
$$+p^{(4)}\left(t^*\right)\frac{\tau^3}{3!} + o(h^3). \quad (26)$$

За рівністю (22) та (26):

$$p'(t) - S'_{4,0}(\overline{p},t) = -\frac{1}{4}p'''(t)h^2 - \frac{1}{4}p^{(4)}(t)h^2\tau -$$
$$-p^{(5)}(t)h^4\left(\frac{161}{5760}h^4 - \frac{7}{48}h^2\tau^2 + \frac{1}{24}\tau^4\right) + o\left(h^3\right).$$

Теорему доведено.

Наслідок 1. Для сплайну $S'_{4,0}(p,t)$, при $p(t) \in C^3$ є вірним

$$\left\|p'(t) - S'_{4,0}(\overline{p},t)\right\|_C \le \frac{h^2}{4}\left\|p'''(t)\right\|_C + \frac{2}{3}\varepsilon\left\|p(t)\right\|_C + o\left(h^3\right).$$

Аналогічно для сплайнів $S''_{4,0}(p,t)$, $S'''_{4,0}(p,t)$, $S^{(4)}_{4,0}(p,t)$.

Теорема 3. Для сплайнів $S''_{4,0}(p,t)$, $S'''_{4,0}(p,t)$, $S^{(4)}_{4,0}(p,t)$ є вірним:

$$\left\|S''_{4,0}(p,t)\right\|_C = \frac{5}{8}\left\|p(t)\right\|_C, \left\|S'''_{4,0}(p,t)\right\|_C = \left\|p(t)\right\|_C,$$

$$\left\|S^{(4)}_{4,0}(p,t)\right\|_C = \left\|p(t)\right\|_C.$$

Доведення є аналогічним доведенню теореми 1.

Теорема 4. Якщо $p(t) \in C^5$, то при $h \to 0$ рівномірно по t має місце асимптотична рівність

$$p''(t) - S''_{4,0}(\overline{p},t) = -\frac{1}{4}p^{(4)}(t)h^2 - p^{(5)}(t)\tau\left(\frac{7}{24}h^2 - \frac{1}{6}\tau^2\right) + o\left(h^2\right),$$

$$p'''(t) - S'''_{4,0}(\overline{p},t) = -p^{(5)}(t)\left(\frac{7}{24}h^2 - \frac{1}{2}\tau^2\right) + o(h).$$

Доведення є аналогічним доведенню теореми 2.

Наслідок 2. Для сплайну $S''_{4,0}(p,t)$ при $p(t) \in C^4$ та для сплайну $S'''_{4,0}(p,t)$ при $p(t) \in C^5$ є вірним:

$$\left\|p''(t) - S''_{4,0}(\overline{p},t)\right\|_C \le \frac{h^2}{4}\left\|p^{(4)}(t)\right\|_C + \frac{5}{8}\varepsilon\left\|p(t)\right\|_C + o\left(h^2\right),$$

$$\left\|p'''(t) - S'''_{4,0}(\overline{p},t)\right\|_C \le \frac{7h^2}{24}\left\|p^{(5)}(t)\right\|_C + \varepsilon\left\|p(t)\right\|_C + o(h).$$

Аналогічно для сплайнів $S'_{4,1}(p,t)$, $S''_{4,1}(p,t)$, $S'''_{4,1}(p,t)$, $S^{(4)}_{4,1}(p,t)$ маємо такі теореми:

Теорема 5. Для сплайнів $S'_{4,1}(p,t)$, $S''_{4,1}(p,t)$, $S'''_{4,1}(p,t)$, $S^{(4)}_{4,1}(p,t)$ є вірним:

$$\left\|S'_{4,1}(p,t)\right\|_C = \left\|p(t)\right\|_C, \qquad \left\|S''_{4,1}(p,t)\right\|_C = \frac{35}{32}\left\|p(t)\right\|_C,$$

$$\left\|S'''_{4,1}(p,t)\right\|_C = 2\left\|p(t)\right\|_C, \qquad \left\|S^{(4)}_{4,1}(p,t)\right\|_C = 2\left\|p(t)\right\|_C.$$

Теорема 6. Якщо $p(t) \in C^5$, то при $h \to 0$ рівномірно по t має місце асимптотична рівність

$$p'(t)-S'_{4,1}(\bar{p},t)=p^{(5)}(t)\left(\frac{319}{5760}h^4-\frac{1}{48}h^2\tau^2+\frac{1}{24}\tau^4\right)+\mathrm{o}\left(h^3\right),$$

$$p''(t)-S''_{4,1}(\bar{p},t)=-p^{(5)}(t)\tau\left(\frac{1}{24}h^2+\frac{1}{6}\tau^2\right)+\mathrm{o}\left(h^2\right),$$

$$p'''(t)-S'''_{4,1}(\bar{p},t)=-p^{(5)}(t)\left(\frac{1}{24}h^2-\frac{1}{2}\tau^2\right)+\mathrm{o}(h).$$

Наслідок 3. Для сплайнів $S'_{4,1}(p,t)$, $S''_{4,1}(p,t)$, $S'''_{4,1}(p,t)$ при $p(t)\in C^5$ є вірним:

$$\left\|p'(t)-S'_{4,1}(\bar{p},t)\right\|_C\le\frac{319h^4}{5760}\left\|p^{(5)}(t)\right\|_C+\varepsilon\left\|p(t)\right\|_C+\mathrm{o}\left(h^3\right),$$

$$\left\|p''(t)-S''_{4,1}(\bar{p},t)\right\|_C\le\frac{h^3}{24}\left\|p^{(5)}(t)\right\|_C+\frac{35}{32}\varepsilon\left\|p(t)\right\|_C+\mathrm{o}\left(h^2\right).$$

$$\left\|p'''(t)-S'''_{4,1}(\bar{p},t)\right\|_C\le\frac{h^2}{12}\left\|p^{(5)}(t)\right\|_C+\varepsilon\left\|p(t)\right\|_C+\mathrm{o}(h).$$

Теорема 7. Для сплайнів $S'_{4,2}(p,t)$, $S''_{4,2}(p,t)$, $S'''_{4,2}(p,t)$, $S^{(4)}_{4,2}(p,t)$ є вірним:

$$\left\|S'_{4,2}(p,t)\right\|_C=1,218\left\|p(t)\right\|_C,\qquad\left\|S''_{4,2}(p,t)\right\|_C=\frac{565}{384}\left\|p(t)\right\|_C,$$

$$\left\|S'''_{4,2}(p,t)\right\|_C=\frac{2683}{960}\left\|p(t)\right\|_C,\qquad\left\|S^{(4)}_{4,2}(p,t)\right\|_C=\frac{43}{15}\left\|p(t)\right\|_C.$$

Теорема 8. Якщо $p(t)\in C^5$, то при $h\to 0$ рівномірно по t має місце асимптотична рівність

$$p'(t)-S'_{4,2}(\bar{p},t)=\frac{p^{(5)}(t)}{24}\left(\frac{7h^4}{240}+\frac{300961h^2\tau^2}{14400}+\tau^4\right)+\mathrm{o}\left(h^3\right),$$

$$p''(t)-S''_{4,2}(\bar{p},t)=\frac{p^{(5)}(t)\tau}{6}\left(\frac{300961h^2}{28800}+\tau^2\right)+\mathrm{o}\left(h^2\right),$$

$$p'''(t)-S'''_{4,2}(\bar{p},t)=\frac{p^{(5)}(t)}{2}\left(\frac{300961h^2}{86400}+\tau^2\right)+\mathrm{o}(h).$$

Наслідок 4. Для сплайнів $S'_{4,2}(p,t)$ та $S''_{4,2}(p,t)$ при $p(t)\in C^3$ є вірним:

$$\left\|p'(t)-S'_{4,2}(\bar{p},t)\right\|_C \le \frac{306241h^4}{1382400}\left\|p^{(5)}(t)\right\|_C + 1,218\varepsilon\left\|p(t)\right\|_C + o\left(h^3\right),$$

$$\left\|p''(t)-S''_{4,2}(\bar{p},t)\right\|_C \le \frac{308161h^3}{345600}\left\|p^{(5)}(t)\right\|_C + \frac{565}{384}\varepsilon\left\|p(t)\right\|_C + o\left(h^2\right),$$

$$\left\|p'''(t)-S'''_{4,2}(\bar{p},t)\right\|_C \le \frac{322561h^2}{172800}\left\|p^{(5)}(t)\right\|_C + \frac{2683}{960}\varepsilon\left\|p(t)\right\|_C + o(h)$$

Наведемо приклади реалізації першої та другої похідних сплайну $S_{4,0}(p,t)$. На графіку (рис.1 (а)) точками відмічено відліки сигналу та подано реалізацію згладжування сплайном $S_{4,0}(p,t)$, похідні $S'_{4,0}(p,t)$, $S''_{4,0}(p,t)$ відображено на рис.1 (б) та (в) відповідно. Пунктиром (вертикальні ствпці на рис. 1 (б, в)) відмічено окіл особливих точок сигналу. Як видно з графіків, в особливих точках сигналу, яким відповідають нулі похідної $S'_{4,0}(p,t)$, можна побачити мінімум або максимум сплайну $S_{4,0}(p,t)$, причому за мінімум відповідає додатньо визначена функція другої похідної $S''_{4,0}(p,t)$ з рис.1 (в), а за максимум – від'ємно визначена функція.

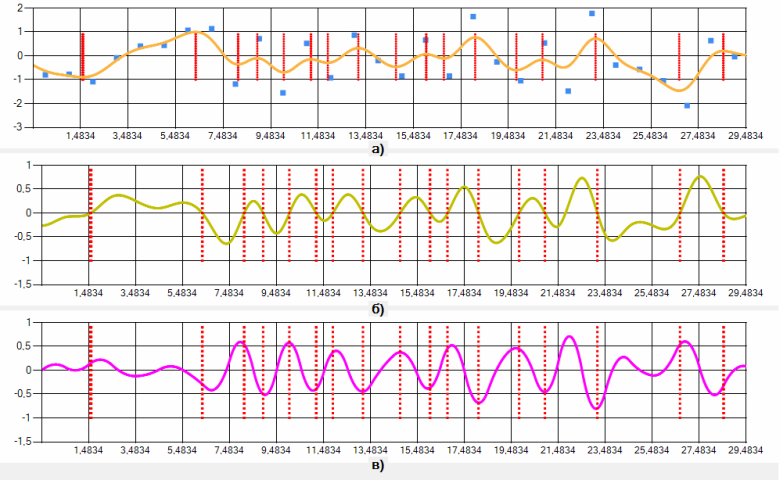


Рисунок 1 – Реалізація аналогового сигналу та згладжування його за допомогою сплайнів: а) відліки сигналу та сплайн $S_{4,0}(p,t)$; б) сплайн $S'_{4,0}(p,t)$; в) сплайн $S''_{4,0}(p,t)$.

Висновки. В роботі отримано явний вигляд похідних поліноміальних сплайнів $S_{4,0}(p,t), S_{4,1}(p,t), S_{4,2}(p,t)$. Подано та доведено теореми про норми та оцінки якості апроксимації функції зазначеними сплайнами, наведено приклад їх застосування для визначення особливих точок випадкового сигналу.

Подальші дослідження можуть полягати в дослідженні похідних поліноміальних сплайнів більш високого порядку задля реалізації їх у задачах обробки цифрових сигналів та узагальненні теоретичних результатів досліджень для двовимірних та багатовимірних випадків.

Бібліографічні посилання

1. Приставка П. О. Лінійні комбінації *B*-сплайнів, близькі до інтерполяційних у середньому, в задачі моделювання аналогових сигналів // Актуальні проблеми автоматизації та інформаційних технологій : зб. наук. праць. Дніпропетровськ. 2011. Т. 15. С. 4–17.

2. Лигун А. А., Шумейко А. А. Асимптотические методы восстановления кривых. Киев. 1996. 358 с.

3. Лигун А. А., Кармазина В. В. О восстановлении эмпирической функции плотности распределения с помощью гистосплайнов второго порядка. Днепродзержинск. 1989. 30 с. Деп. в УкрНИИНТИ 8.06.89. №1559– Ук89.

4. Лигун А. А., Кармазина В. В. Восстановление функций плотности распределения и их производных с помощью кубических гистосплайнов. Днепродзержинск. 1989. 38 с. Деп. в УкрНИИНТИ 13.11.89. №2569– Ук89.

5. Приставка П. О. Поліномаільні сплайни при обробці даних. Дніпропетровськ. 2004. 236 с.

Надійшла до редколегії 07.09.16.

УДК 519.245; 519.674; 004.032.24:004.272

П.О. Приставка, А.К. Шевченко

Національний авіаційний університет

ДОСЛІДЖЕННЯ РЕАЛІЗАЦІЇ ЛІНІЙНОГО ОПЕРАТОРА ЗГОРТКИ ЦИФРОВОГО ЗОБРАЖЕННЯ ПРИ 16-БІТНИХ ОБЧИСЛЕННЯХ

Досліджено реалізацію згортки складової растра з квадратною маскою лінійного двовимірного фільтра. Доведено, що за допомогою 16-бітних операцій та трансляції мовою асемблера можна отримувати виграш від 8 до 11 разів по швидкодії, при використанні у програмному забезпеченні.

Ключові слова: *згортка, цифрове зображення, лінійний оператор, авто-векторизація , SIMD, асемблер.*

Исследована реализация свертки составляющей растра цифрового изображения с квадратной маской линейного двумерного фильтра. Доказано, что при использовании 16-битных операций и встроенного ассемблера, можно получить выигрыш от 8 до 11 раз, по быстродействию в программном обеспечении.

Ключевык слова: *свертка, цифровое изображение, линейный оператор, авто-векторизация , SIMD, ассемблер.*

It has been researched the convolution of digital image raster component and two-dimensional linear filter square mask. It has been proved one may obtain software performance improvement from 8 to 11 times in the case of 16-bit SIMD operation and GCC Inline Assembly usage.

Keywords*: convolution, digital images, linear operator, auto-vectorization, SIMD, assembler.*

Постановка проблеми. Процедура автоматичної векторизації програмного коду, базис якої складають SIMD команди асемблера, є основою для прискорення значної кількості програм [1].

Отже, стан розвитку сучасних компіляторів можна відслідковувати за тим, як кожен із них справляється із завданням векторизації коду, та чи є після цього помилки виконання (що не є рідкістю). Сучасні спеціалісти надають цьому чиннику дуже великого значення, отже, використовують саме той компілятор, що призведе до підвищення швидкодії їх програм.

Особливо важливим є прискорення процесу обробки цифрових зображень (ЦЗ) як окремо, так і відеокадрів (при опрацюванні відеопотоку).

Тут треба враховувати три головні фактори: 1) обчислювальна складність методу обробки ЦЗ; 2) чи оптимально він (метод) був реалізований, у вигляді програмного коду; 3) та апаратні можливості.

Отже, надамо приклад: однією з найпростіших операцій при роботі з ЦЗ є згортка кольорової (або монохромної) складової растра з квадратною маскою деякого лінійного двовимірного фільтра:

$$p_{i,j} = F\left(p_{i,j}\right) = \sum_{ii=i-r_i}^{i+r_i} \sum_{jj=j-r_j}^{j+r_j} \gamma_{ii-i,jj-j}\, p^{o}_{ii,jj} \qquad (1)$$

де: $i = \overline{-\frac{k_i}{2}, \frac{k_i}{2}}$; $j = \overline{-\frac{k_j}{2}, \frac{k_j}{2}}$; а $p_{i,j}$ – кольорова складова растра; $F\left(i,j\right)$ – лінійні фільтрації зображення; $\left(i,j\right)$ – індекс растра пікселів; k_i, k_j – розміри кадру зображення; $p^{o}_{ii,jj}$ – кольорова складова растра ідеального неспотвореного зображення; $\gamma_{ii-i,jj-j}$ – елемент маски фільтра; $\left(2r_i+1\right)\times\left(2r_j+1\right)$ – розмір маски фільтра.

Актуальним є, на прикладі реалізації даної операції, провести дослідження швидкодії її виконання, з урахуванням можливостей оптимізації коду та вибору математичного забезпечення.

Аналіз публікацій та постановка задачі. Сучасне програмне забезпечення залежить від якості інструментів, за допомогою яких воно було створене – компіляторів та інтерпретаторів. Від того, на скільки оптимально компілятор векторизує отриманий код, залежить, чи буде програма, за критерієм швидкодії, задовольняти користувача. В даному контексті – швидкодія залежить від можливостей компілятора векторизувати код програми та від апаратної складової.

На сьогоднішній день компілятори GNU Compiler Collection (далі GCC/G++) [2] та Clang [3] є незамінними інструментами будь-якого розробника. Це заслуга багаторічної праці як відкритого співтовариства FSF (Free Software Foundation) [4], так і університетської розробки, за великої підтримки фірми Apple (на початку). Спільнота FSF розробляє і всіляко підтримує GCC, а Apple зробила ставку на Clang як на компілятор, який є базовим для усіх її продуктів.

GCC являє собою набір компіляторів для різних мов програмування, розроблений в рамках проекту GNU. Початок GCC було покладено Річардом Столлманом, який реалізував перший варіант GCC у 1985 році на нестандартному і непереносному діалекті мови Паскаль; пізніше компілятор був переписаний (Леонардом Тауером і Річардом Столлманом) за допомогою мови C, та випущений у 1987 році як компілятор проекту GNU.

Clang є фронтендом для мов програмування C, C++, Objective-C, Objective-C++ та OpenCL. Для оптимізації (векторизації) і кодогенерації у ньому використовується фреймворк LLVM. Метою проекту є створення заміни GCC. Розробка ведеться згідно з концепцією opensource. У проекті беруть участь працівники декількох корпорацій, у тому числі Google та Apple. Вихідний код доступний на умовах bsd-подібної ліцензії.

Отже, GCC та Clang використовують для отримання оптимального вихідного програмного забезпечення за показником швидкодії виконання. Під оптимальністю розуміється те, що використання команд асемблера SIMD буде на ділянках програми, що представляють найбільш ресурсомісткі/ресурсозатратні частини програмного продукту (ПП), найбільш поширеним. Під ресурсомісткістю слід розуміти те, що дана ділянка програми найбільш часто використовується або містить у собі велику вкладеність циклів, що і призводить до уповільнення загальної програми і т.ін. Таким чином, використовуючи команди SIMD-асемблера, кінцевий ПП отримує приріст продуктивності. Але слід відзначити той автоматизм, з яким компілятори працюють (GCC та Clang): яким би не було завдання, оптимізація йде приблизно за таким алгоритмом перетворення типів даних:

$$\{u8\} \rightarrow \{u16\} \rightarrow \{u32\} \rightarrow \{f32\}[\ldots],$$
(найбільш ресурсномістка частина операцій з f32): […]
$$\{f32\} \rightarrow \{u32\} \rightarrow \{u16\} \rightarrow \{u8\}.$$

Операцію з f32 ([…]) слід розуміти як неефективну (що знижує продуктивність), але таку, що призводить до точності «біт у біт». З цього випливає, що деякі ділянки програмного коду слід переписати вручну і тут вступають у дію фахівці зі знанням мови асемблера (як базового рівня, так і SIMD-команд), які й відповідальні за ухвалення рішення про дотримання принципу «біт у біт», і/або переписування коду (із застосуванням цілочисельного обчислення). У підсумку програма набуває вигляду, який трохи відштовхує середньостатистичного фахівця у сфері програмування, але дає приріст, як мінімум, у 4 рази. Мало відомий факт, що у таких великих

корпораціях, як Google та Apple, є штат програмістів, які розробляють нові чисельні методи, що використовують “single point” арифметику, але результати (за критерієм точності) подібні до обчислень з використанням “floating point” арифметики. Наслідком використання таких алгоритмів є мінімальна припустима втрати точності на противагу до значного підвищення швидкодії алгоритму. Цей ПП роблять за допомогою найновішого асемблера, застосовуючи його найновіші SIMD-команди та, найчастіше, це інлайн асемблер (Inline Assembly).

Тож у подальшому викладенні поставимо за мету дослідити переваги використання оптимізованого коду під час реалізації операції згортки (1) при обробці ЦЗ та отримати нові лінійні оператори для забезпечення виконання саме такої обчислювальної операції.

Виклад основного матеріалу. Десять років тому було домінування CISC-архітектури процесорів. Проте, з розвитком мобільного сегмента ринку (мобільні телефони, планшети, WiFi тощо) першість перейшла до RISC-архітектури процесорів, а точніше – до стандартів процесорів фірми ARM. Найбільш популярними ядрами процесорів з 32-bit архітектурою стали: Cortex-A8, Cortex-A9 та Cortex-A15 [5, с. 13]. Наприкінці 2011 року було опубліковано нову версію 64-bit архитектури ARMv8 з такими стандартами ядер процесорів: Cortex-A53 та Cortex-A57. Найбільш популярним ядром для сегмента одноплатних персональних комп’ютерів (далі – SBC) стала версія Cortex-A53 [6, с. 13]. Прикладом SBC можуть слугувати DragonBoard™ 410c та Raspberry Pi 3 [7].

Векторизація коду дозволяє виконувати, за один такт процесора, безліч одноманітних дій з банком даних (векторним регістром) – SIMD-операцію. На вхід будь-якої SIMD-операції подаються декілька векторних регістрів, довжина яких варіюється залежно від архітектури процесора (CISC або RISC): у CISC довжина векторного регістру може складати 128-bit (xmm) або 256-bit (ymm), 512-bit (zmm); для RISC ця довжина завжди фіксована та складає 128-bit, незалежно від розрядності процесора [5, с. 13; 6, с. 13].

Прикладом, що демонструє якість процедури авто-векторизації програмного забезпечення, з використанням компілятора “GNU C++ compiler” (далі – g++), для архітектур ARMv7-A (прапори компіляції: -O3 -fstrict-aliasing fauto-inc-dec -fprefetch-loop-arrays -mfpu=neon) може бути дослідження її швидкодії на операції (1). Швидкодія програми, що реалізувала операцію (1), досліджувалася на операційній системі Android 5.0.1 з архітектурою процесора Cortex-A53 (процесор: Mediatek MT6752), що дозволяє оцінити як якість авто-векторизації,

так і зворотну сумісність ARMv7-A коду програми з ARMv8 апаратною складовою.

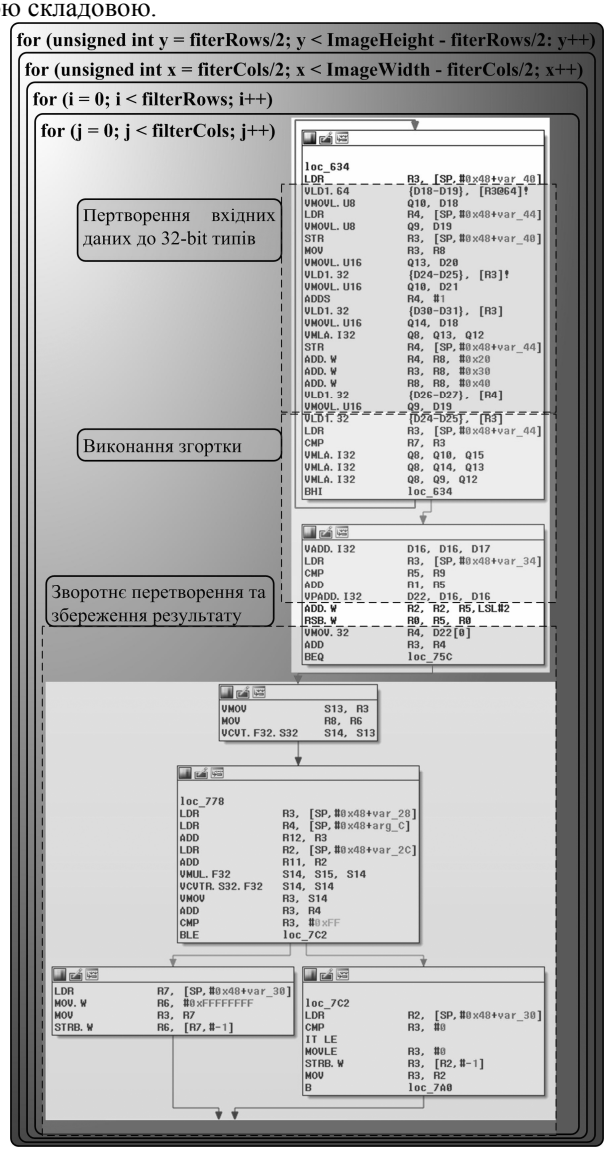


Рисунок 1 – Приклад авто-векторизації операції згортки, отриманий за допомогою програми IDA Pro (1)

Результат авто-векторизації операції (1) представлено на (рис. 1), де подано частину програми мовою асемблера (в основному, SIMD фірми ARM – NEON (32-бітна версія)), яку умовно поділено на 3 секції: 1) перетворення вхідних даних до 32-bit типів (таких як int/float); 2) виконання основного алгоритму; 3) зворотне перетворення та збереження результату (трансляція із 32-bit типів (int/float) у необхідний вихідний тип даних (у наведеному прикладі – це unsigned char (8-bit)).

Отже, як видно з діаграми, для виконання згортки попередньо відбувається перетворення даних до 32-bit типів (рис. 2).

```
VLD1.64         {D18-D19}, [R3@64]!
VMOVL.U8        Q10, D18
VMOVL.U8        Q9, D19
VMOVL.U16       Q13, D20
VMOVL.U16       Q10, D21
VMOVL.U16       Q14, D18
VMOVL.U16       Q9, D19
VMLA.I32        Q8, Q10, Q15
VMLA.I32        Q8, Q14, Q13
VMLA.I32        Q8, Q9, Q12
VADD.I32        D16, D16, D17
VPADD.I32       D22, D16, D16
VMOV.32         R4, D22[0]
```

Рисунок 2 – Приклад автоматичного перетворення компілятором даних до 32-bit типу

Варто також зазначити, що в регістрі Q9 (D18, D19) було передано 16 по 8-bit значень інтесивності кольорової складової растра ЦЗ (ширина векторного регістру в Q9 складає 128-bit = 16 x 8-bit), які були трансльовані у 16 по 32-bit значень та розміщені в регістрах Q9, Q10, Q13, Q14.

Узагальнено, векторізація пройшла таким чином: розгортання внутрішнього циклу, кроком по 4 елемента (та операцій із зображенням/ядром згортки) за одну ітерацію цикла. Це мало б призвести до прискорення операціїї згортки на розмірах ядра, більших за 4 (але не дуже суттево, у порівнянні з експертною векторизацією). Це припущення демонструє рис. 3.

Такий вид авто-векторизації є достатньо поширеним і для перевірки самої авто-векторизації компілятора g++ наведена програма була скомпільована іншим компілятором, що входить у пакет nfk r10e—rc4 (далі – NDK) – LLVM [3]. Результат виявився тим самим, за винятком інших назв регістрів, проте, послідовність дій була еквівалентною (рис.1).

В секції «Виконання згортки» (рис. 1) усі операції відбуваються з 32-bit типами (як i32, так і f32), що дозволяє розробникам ПП, котрі використовують оптимізацію/авто-векторизацію (за допомогою активації прапора -O3 або -Ofast) розраховувати, що вона відбудеться з високою точністю та не призведе до втрати точності розрахунків.

Проте, якщо вхідні дані транслювати в 16-bit діапазон, то, як виявляється, виконання алгоритму операції (1) буде займати, як мінімум, у 4 рази менше команд, а крок зміниться на 32 елемента за один раз – незалежно від розміру ядра згортки. Для підтвердження такого зауваження було проведено серію експериментів, під час яких порівнювалася швидкодія операції (1) для різних розмірів маски фільтра (від 3x3 до 11x11) та різних розмірах ЦЗ (від 10^6 до $8 \cdot 10^6$ пікселів).

Для кожного розміру маски фільтра та для кожного розміру ЦЗ було проведено по 1000 експериментів для порівняння швидкодії при авто-вектризації коду, написаного мовою C++, та при трансляції вручну типів у 16-бітний давпазон мовою асемблера. Усереднене значення T відношення часу виконання:

$$T = \frac{час\ виконання\ при\ трансляції\ 16-bit}{час\ виконання\ при\ авто-векторизації}$$

подано на графіку (рис. 3). Як видно, виграш у часі при використанні 16-бітних обчислень складає від 8 до 11 разів у порівнянні з часом виконання операції (1) при авто-векторизації компілятором.

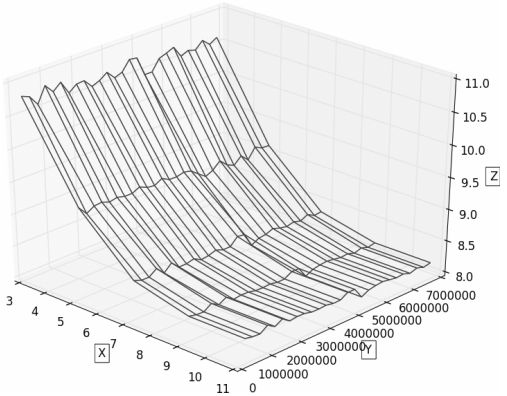


Рисунок 3 – Відношення часу виконання при 16-бітних обчисленнях до часу виконання операції (1) при авто-векторизації: [x] – розмір ядра згортки; [y] – розмір зображення у пікселях; [z] – прискорення

Проте, варто зауважити про існуючі обмеження на використання 16-бітних обчислень при реалізації згортки (1). З урахуванням того, що $p^{\circ}_{ii,jj}$ (значення кольорової складової растра) цілочисельне з діапазону від 0 до 255, можемо записати:

$$\left| \sum_{ii=i-r_i}^{i+r_i} \sum_{jj=j-r_j}^{j+r_j} \gamma_{ii-i,jj-j} p^{\circ}_{ii,jj} \right| \le 255 \cdot \sum_{ii=i-r_i}^{i+r_i} \sum_{jj=j-r_j}^{j+r_j} \left| \gamma_{ii-i,jj-j} \right| .$$

Отже, для забезпечення можливості виконання 16-бітної операції достатньо вимагати, щоб елементи маски γ були типу ShortInt та при реалізації виразу (1) виконувалася нерівність

$$\sum_{ii=i-r_i}^{i+r_i} \sum_{jj=j-r_j}^{j+r_j} \left| \gamma_{ii-i,jj-j} \right| \le 127 . \qquad (2)$$

У подальшому викладенні подамо приклади масок, що задовольняють умові (2). Не зменшуючи загальності, нехай задано деякий безрозмірний растр, кожному пікселю якого поставлено у відповідність двійку індексів $\{(i,j)\}_{i,j\in\square}$, що визначають його місцеположення. Позначимо $\{p_{i,j}\}_{i,j\in\square}$ – для запису обчислювальної схеми при роботі з послідовностями кольорових складових (червоною, зеленою та синьою).

Для реалізації субполосної фільтрації послідовності $\{p_{i,j}\}_{i,j\in\square}$ припускаємо, що значення інтенсивності в кожному пікселі можна подати у вигляді суми

$$p_{i,j} = pL_{i,j} + pH_{i,j},$$

де $pL_{i,j}$, $pH_{i,j}$ – відповідно низькочастотні/високочастотні складові, які можна визначати на основі таких лінійних операторів [8]:

$$pL_{i,j} = L\left(p^{i,j}\right) = \sum_{ii=i-1}^{i+1} \sum_{jj=j-1}^{j+1} \gamma L_{ii-i,jj-j} p_{ii,jj} \ , \ i,j \in \mathbf{Z},$$

$$pH_{i,j} = H\left(p^{i,j}\right) = \sum_{ii=i-1}^{i+1} \sum_{jj=j-1}^{j+1} \gamma H_{ii-i,jj-j} p_{ii,jj} \ , \ i,j \in \mathbf{Z},$$

де

$$\gamma L = \frac{1}{64}\begin{pmatrix} 1 & 6 & 1 \\ 6 & 36 & 6 \\ 1 & 6 & 1 \end{pmatrix};\ \gamma H = \frac{1}{64}\begin{pmatrix} -1 & -6 & -1 \\ -6 & 28 & -6 \\ -1 & -6 & -1 \end{pmatrix} \qquad (3)$$

або

$$\gamma L = \frac{1}{36}\begin{pmatrix} 1 & 4 & 1 \\ 4 & 16 & 4 \\ 1 & 4 & 1 \end{pmatrix};\ \gamma H = \frac{1}{36}\begin{pmatrix} -1 & -4 & -1 \\ -4 & 20 & -4 \\ -1 & -4 & -1 \end{pmatrix}. \qquad (4)$$

Зауважимо, що тут і далі в роботі виконання умови (2) вимагається для цілочисельних елементів масок згорток (3) – (4) до ділення на величину, винесену в знаменники.

Для контрастування зображень можна використовувати оператор

$$pK_{i,j} = K\left(p^{i,j}\right) = \sum_{ii=i-2}^{i+2}\sum_{jj=j-2}^{j+2} \gamma K_{ii-i,jj-j}\, p_{ii,jj}\ ,\ i,j \in \mathbf{Z},$$

де

$\left\{pK_{i,j}\right\}_{i,j\in\mathbf{Z}}$ – кольорова складова відконтрастованого растра;

$$\gamma K = \frac{1}{16}\begin{pmatrix} 0 & 0 & 2 & 0 & 0 \\ 0 & 3 & -13 & 3 & 0 \\ 2 & -13 & 48 & -13 & 2 \\ 0 & 3 & -13 & 3 & 0 \\ 0 & 0 & 2 & 0 & 0 \end{pmatrix}, \qquad (5)$$

який, до того ж, є псевдозворотним оператором [9] до оператора $L\left(p^{i,j}\right)$ низькочастотної фільтрації з маскою (4).

Стабілізація цифрового зображення, спотвореного мікрорухом камери фіксації (підвищення різкості) [10], можлива за використання такого оператора:

$$pR_{i,j} = R\left(p^{i,j}\right) = \sum_{ii=i-2}^{i+2}\sum_{jj=j-2}^{j+2} \gamma R_{ii-i,jj-j}\, p_{ii,jj}\ ,\ i,j \in \mathbf{Z},$$

де $\left\{pR_{i,j}\right\}_{i,j\in\mathbf{Z}}$ – кольорова складова растра з покращеною різкістю;

$$\gamma R = \frac{1}{42}\begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & -8 & 1 & 0 \\ -1 & -8 & 74 & -8 & -1 \\ 0 & 1 & -8 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}, \qquad (6)$$

або

$$\gamma R = \frac{1}{15}\begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & -4 & 1 & 0 \\ -1 & -4 & 31 & -4 & -1 \\ 0 & 1 & -4 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}. \qquad (7)$$

Зауважимо, що використання масок (6) та (7) забезпечує, відповідно, менше та більше підвищення різкості.

Для чотирикратного рекурентного поповнення кількості членів послідовності $\left\{p_{i,j,0}\right\}_{i,j\in\mathbf{Z}}$ (двократного рекурентного збільшення лінійних розмірів цифрового зображення) [11; 12] достатньо на кожному κ-му ($\kappa = 1,2,\ldots$) кроці рекурсії визначати члени нової послідовності кольорових складових растра $\left\{p_{2i,2j,\kappa}\right\}_{i,j\in\mathbf{Z}}$ на основі лінійних операторів, що основані на даних попереднього кроку рекурсії:

$$p_{2i,2j,\kappa} = A\left(p^{\kappa-1,i,j}\right), \qquad p_{2i+1,2j,\kappa} = B\left(p^{\kappa-1,i,j}\right),$$

$$p_{2i,2j+1,\kappa} = C\left(p^{\kappa-1,i,j}\right), \qquad p_{2i+1,2j+1,\kappa} = D\left(p^{\kappa-1,i,j}\right),$$

$$i,j \in \mathbf{Z},$$

де $A\left(p^{\kappa-1,i,j}\right)$; $B\left(p^{\kappa-1,i,j}\right)$; $C\left(p^{\kappa-1,i,j}\right)$; $D\left(p^{\kappa-1,i,j}\right)$ можна задавати з умов масштабування. Так, якщо є потреба збільшувати зображення зі згладжуванням, то мають місце такі оператори:

$$p_{a,b,\kappa} = \sum_{ii=i-1}^{ii+1}\sum_{jj=j-1}^{j+1} \gamma\Lambda_{ii-i,jj-j}\, p_{ii,jj,\kappa-1},$$

де

$$\Lambda = \begin{cases} A : a = 2i, b = 2j, \\ B : a = 2i+1, b = 2j, \\ C : a = 2i, b = 2j+1, \\ D : a = 2i+1, b = 2j+1; \end{cases} \tag{8}$$

$$\gamma A = \frac{1}{64}\begin{pmatrix} 1 & 6 & 1 \\ 6 & 36 & 6 \\ 1 & 6 & 1 \end{pmatrix}; \qquad \gamma B = \frac{1}{16}\begin{pmatrix} 0 & 0 & 0 \\ 1 & 6 & 1 \\ 1 & 6 & 1 \end{pmatrix};$$

$$\gamma C = \frac{1}{16}\begin{pmatrix} 0 & 1 & 1 \\ 0 & 6 & 6 \\ 0 & 1 & 1 \end{pmatrix}; \qquad \gamma D = \frac{1}{4}\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

Коли немає потреби збільшувати зі згладжуванням, то варто застосовувати такі оператори:

$$p_{a,b,\kappa} = \sum_{ii=i-1}^{ii+1} \sum_{jj=j-1}^{j+1} \gamma\Lambda_{ii-i,jj-j}\, p_{ii,jj,\kappa-1},$$

де Λ – визначається згідно з (6);

$$\gamma A = \frac{1}{50}\begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ -1 & 2 & 46 & 2 & -1 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}; \quad \gamma B = \frac{1}{82}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -7 & 0 & 0 \\ -1 & 2 & 46 & 2 & -1 \\ -1 & 2 & 46 & 2 & -1 \\ 0 & 0 & -7 & 0 & 0 \end{pmatrix};$$

$$\gamma C = \frac{1}{82}\begin{pmatrix} 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 2 & 2 & 0 \\ 0 & -7 & 46 & 46 & -7 \\ 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & -1 & -1 & 0 \end{pmatrix}; \quad \gamma D = \frac{1}{20}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 \\ 0 & -1 & 7 & 7 & -1 \\ 0 & -1 & 7 & 7 & -1 \\ 0 & 0 & -1 & -1 & 0 \end{pmatrix}.$$

Для рекурентного двократного збільшення масштабу (двократного зменшення горизонтального та вертикального розмірів) зображення необхідно на кожному κ-му ($\kappa = 1,2,\ldots$) кроці рекурсії чотирикратно зменшувати кількість пікселів, звільняючи у новому κ-му растрі місце з-під трьох пікселів: праворуч, зверху та зверху-навскіс від кожного (i,j)-го пікселя $(\kappa-1)$-го растра. Тобто, якщо $\{p_{i,j,\kappa}\}_{i,j\in\mathbf{Z}}$ – послідовність однієї із кольорових складових κ-го зменшеного растра,

то $p_{i,j,\kappa} = p_{2i,2j,\kappa-1}$, при цьому пам'ять під розміщення величин $p_{2i+1,2j,\kappa-1}$, $p_{2i,2j+1,\kappa-1}$, $p_{2i+1,2j+1,\kappa-1}$ може бути вивільнена. Але, окрім тривіального визначення членів послідовності $\left\{p_{i,j,\kappa}\right\}_{i,j\in\mathbf{Z}}$ згідно з (7), реалізують збільшення масштабу зображення зі згладжуванням, контрастуванням, направленою фільтрацією тощо, залежно від конкретних потреб. У такому разі величини $p_{i,j,\kappa}$ визначаються на підставі деякого лінійного функціоналу:

$$p_{i,j,\kappa} = F\left(p^{\kappa-1,2i,2j}\right) = \sum_{ii=2i-r}^{2i+r}\sum_{jj=2j-r}^{2j+r} \gamma_{ii-i,jj-j}\, p_{ii,jj,\kappa-1}$$

$i, j \in \mathbf{Z}$, $r = 1,2,\ldots$, що побудований на даних попереднього кроку рекурсії, а в якості γ можна використовувати маски (3) – (7).

Висновки. У роботі проведено дослідження реалізації згортки монохромної складової растра цифрового зображення з квадратною маскою лінійного двовимірного фільтра. Експериментально доведено, що при авто-векторизації коду, що реалізує операцію (1), за допомогою компілятора gcc/g++ для архітектури ARMv7-A, відбувається автоматична конвертація типів даних для 32-розрядних обчислень, задля забезпечення гарантованої точності обчислень. Проте, проведені авторами дослідження доводять, що за допомогою лише 16-бітних операцій та трансляції мовою асемблера ресурсномісткої частини алгоритму можна отримувати виграш від 8 до 11 разів по швидкодії при виконанні операції (1). В роботі наведено умову (2), за якою є можливим досягнення такого виграшу, та подано приклади відповідних лінійних операторів, що можуть мати реалізацію при розробці програмного забезпечення щодо обробки цифрових зображень (та відео).

Подальші дослідження мають за мету: проведення оцінки виграшу при застосуванні 16-бітних обчислень для архітектури процесорів ARMv8 (Mediatek MT6752), що реалізує операцію (1), та отримання нових відповідних лінійних операторів, наприклад, для визначення особливостей ЦЗ, масштабного аналізу тощо.

Бібліографічні посилання

1. Flynn M. J. Very high speed computers. // Proceedings of the IEEE: [Vol: 54. Issue: 12]. 1966. P. 1901–1909.

2. Гриффитс А. GCC. Настольная книга пользователей, программистов и системных администраторов [пер. з англ.]. Киев.

2004. 624 с.

3. Lopes B. C., Auler R. Getting Started with LLVM Core Libraries. Kiyiv. 2014. 314 с.

4. Gay J. Free Software, Free Society: Selected Essays of Richard M. Stallman. Kiyiv. 2002. 230 с.

5. ARM® Cortex®-A15 MPCore™ Processor: (технічний довідник з архітектури процесорів серії Cortex-A15 фірми ARM). 2011–2013. С 392. URL: http://infocenter.arm.com/help/topic/com.arm.doc.ddi0438i/ DDI0438I_cortex_a15_r4p0_trm.pdf. Назва з екрана.

6. ARM® Cortex®-A53 MPCore Processor: (технічний довідник з архітектури процесорів серії Cortex-A53 фірми ARM). 2013–2014. С. 620. URL: http://infocenter.arm.com/help/topic/com.arm.doc.ddi0500g/ DDI0500G_cortex_a53_trm.pdf. – Назва з екрана.

7. Raspberry Pi 3 Model B: (технічна специфікація що містить опис апаратного забезпечення одноплатного персонального комп'ютера – Raspberry Pi 3 Model B). 2015. С. 1. URL: https://cdn.sparkfun.com/datasheets/Dev/RaspberryPi/2020826.pdf. – Назва з екрана.

8. Приставка П. О. Обчислювальні аспекти застосування поліноміальних сплайнів при побудові фільтрів // Актуальні проблеми автоматизації та інформаційних технологій : зб. наук. праць. Дніпропетровськ. 2006. Т. 10. С. 3–14.

9. Приставка П. О. Побудова контрастних фільтрів за використанням поліноміальних сплайнів // Актуальні проблеми автоматизації та інформаційних технологій : зб. наук. праць. Дніпропетровськ. 2007. Т. 11. С. 15–22.

10. Приставка П. О., Чолишкіна О. Г. Дослідження комбінованих фільтрів для підвищення різкості зображень // Актуальні проблеми автоматизації та інформаційних технологій : зб. наук. праць. Дніпропетровськ. 2009. Т. 13. С. 39–53.

11. Приставка П. О. Поповнення послідовностей відліків функцій двох змінних на основі поліноміальних сплайнів // Вісник НАУ. 2007. № 3–4. С. 36–39.

12. Приставка П. О. Поповнення зі згладжуванням послідовностей відліків функцій двох змінних на основі сплайнів // Математичне моделювання. 2008. № 1(18). С. 9–12.

Надійшла до редколегії 14.11.16.

УДК 519.163

Ю. С. Шахновский, П. В. Кузнецов

Национальный технический университет «Харьковский политехнический институт»

ОПТИМИЗАЦИЯ СИСТЕМЫ ФИЛЬТРАЦИИ ПАКЕТОВ ПУТЕМ ИЗМЕНЕНИЯ ПОРЯДКА ПРАВИЛ

Рассмотрено решение задачи оптимизации систем фильтрации пакетов сетевых шлюзов с использованием изменения порядка правил, а также графовой модели упорядочивания правил фильтрации. В результате исследования получены точные и эвристические алгоритмы оптимизации порядка правил фильтрации пакетов. Дана оценка качества полученных эвристических алгоритмов.

Ключевые слова: *системы фильтрации пакетов, именение порядка правил, сетевые шлюзы, эвристические алгоритмы.*

Розглянуто рішення задачі оптимізації систем фільтрації пакетів мережевих шлюзів з використанням зміни порядку правил, а також графової моделі упорядкування правил фільтрації. В результаті дослідження отримано точні і евристичні алгоритми оптимізації порядку правил фільтрації пакетів. Надано оцінку якості отриманих евристичних алгоритмів.

Ключові слова: *системи фільтрації пакетів, зміна порядку правил, мережеві шлюзи, евристичні алгоритми.*

The article discusses the problem of optimizing packet filtration systems of network gateways using the reordering rules, and also using the graph model for ordering filtering rules. The study produced the exact and heuristic algorithms to optimize the order of the rules in the packet filtration system. Evaluation of the quality of the obtained heuristic algorithms is provided.

Keywords: *packet filtering systems, change the rules of order, network gateways, heuristic algorithms.*

Постановка проблемы. Системы фильтрации пакетов используются в сетевых шлюзах для задания правил, какие из пакетов пропускать сквозь шлюз, а какие нет. Алгоритм работы такой системы состоит в следующем. Имеется заданная системным администратором последовательность правил.

Каждое правило задает некоторое множество пакетов, которые подчиняются ему. Пакет проверяется на соответствие с правилами, заданными в последовательности от начала к концу. Если при очередном сравнении пакет соответствует правилу, то он не обрабатывается оставшимися правилами последовательности. В противном случае он направляется на проверку соответствия следующему правилу. Среднее время прохождения пакетов через систему будет зависеть от порядка расположения правил.

Анализ последних исследований и публикаций. При оптимизации систем фильтрации стандартный подход состоит в изменении внутреннего содержания правил, что позволяет производить расчеты по одному из этих правил быстрей. При этом не учитываются возможности, которые дает простое изменение порядка применения правил. Примером такого подхода может служить [1]. В статье [2] сформулирована задача оптимизации с использованием изменения порядка правил, и доказана невозможность найти оптимальное решение за полиномиальное время. Поэтому предложено использование эвристических алгоритмов. Предложенный в этой статье алгоритм не использует всех возможностей оптимизации, которые заложены в моделировании оптимизируемой системы с помощью графа.

Рассматриваемое в этой статье решение задачи развивает алгоритм, предложенный в [3].

Нерешенные ранее части общей проблемы. Предыдущие работы не использовали графовую модель упорядочивания правил фильтрации.

Цель статьи заключается в получении точных и эвристических алгоритмов оптимизации порядка правил в системе фильтрации пакетов и оценке качества полученных эвристических алгоритмов.

Изложение основного материала. В системах фильтрации пакетов постоянно ведется подсчет числа пакетов, проходящих через различные правила. Цель этого сбора статистики – определить, какую часть трафика должен оплатить каждый из пользователей. На основании этой информации за некоторый интервал времени можно рассчитать относительную частоту прихода пакетов, соответствующих различным правилам. Разумно предположить, что на следующем интервале времени частоты пакетов изменятся не сильно.

Кроме относительных частот срабатывания каждого правила, для решения задачи оптимизации необходимо знать время расчета по каждому из правил. Это время зависит от применяемых для расчета этих правил программ и может варьироваться в зависимости от типа

ЭВМ, на которой установлена фильтрующая программа. Часто встречаются случаи, когда разницей во времени обработки пакетов разными правилами можно пренебречь, и тогда эти времена можно считать единичными.

Необходимо учитывать ограничения на переупорядочивание правил, связанные с тем, что некоторые пары правил описывают пересекающиеся множества пакетов. В случае изменения порядка двух правил с непустым множеством пресечения, пакеты, принадлежащие пересечению, будут обрабатываться не тем правилом, каким они обрабатывались в начальной цепочке, т.е. не так, как задумал автор правил. Чтобы не допустить этого, на множестве правил можно построить отношение частичного предшествования, в котором задан порядок любой пары правил, описывающих пересекающиеся множества пакетов. Такое частичное упорядочивание удобно представлять в виде ациклического ориентированного графа. Вершинами этого графа будут правила, а дуга из вершины i в вершину j идет тогда и только тогда, если i предшествует j в начальной перестановке и множество пакетов, отвечающих i и j, пересекаются. Такой граф мы будем в дальнейшем называть "графом зависимостей", или ГЗ. Заметим, что поскольку ГЗ является графом, отображающим отношение предшествования, то это отношение не изменится как при транзитивном замыкании, так и при транзитивной редукции графа. Алгоритмы, исследуемые далее, лучше работают, если граф подвергнуть транзитивной редукции на этапе построения. Описание алгоритма транзитивной редукции можно найти в [4].

Для решения сформулированной технической задачи нужно уметь решать следующую математическую задачу:

Известны:

1. Относительная частота встречи пакетов, относящихся к каждому из правил v[i].
2. Время расчета каждого из правил t[i].
3. Граф зависимостей, задающий ограничения на переупорядочивание правил ГЗ.

По заданному порядку правил можно рассчитать суммарное время прохождения всех пакетов через систему: v[1]*t[1]+v[2]*(t[1]+t[2])+…+v[n]*(t[1]+t[2]+…+t[n]).

Эту величину необходимо минимизировать.

Существуют и другие технические задачи, отвечающие данной оптимизационной модели. Примером другой проблемы, где возникает такая постановка задачи, является задача переупорядочивания записей таблицы маршрутизации.

При отсутствии ограничений на перестановку правил оптимальное решение находится с помощью приоритета, когда раньше включаются правила с большим отношением v[i]/t[i].

Утверждение 1: Если ГЗ пуст (т.е. ГЗ не содержит дуг), то оптимум можно получить с помощью приоритетного правила – первым в решение включать ту вершину, у которой v[i]/t[i] максимально.

Доказательство: пусть есть оптимальное решение, для которого это не выполняется. Тогда в нем есть пара вершин k и k+1, что v[k]/t[k]<v[k+1]/t[k+1]. Если поменять эти две вершины местами, время прохождения пакетов, не отвечающих этим вершинам, не изменится. Время прохождения пакетов, отвечающих этим вершинам, будет состоять из двух частей – части, связанной с прохождением предшествующих вершин цепочки, и части, связанной с прохождением этих двух вершин. Первая составляющая не изменится. Вторая составляющая уменьшится для пакетов вершины k+1 на t[k], что даст уменьшение общей оценки на v[k+1]*t[k], а время прохождения пакетов вершины k увеличится на t[k+1], что даст увеличение общей оценки на t[k+1]*v[k]. Общее изменение оценки будет v[k]*t[k+1]-v[k+1]*t[k]. Так как v[k]/t[k]<v[k+1]/t[k+1], то общее время оценки уменьшится. Поэтому начальное решение не было оптимальным.

Этот прием используется системными администраторами шлюзов при ручной оптимизации.

Если ГЗ содержит дуги, то правило предыдущего раздела не дает оптимума, так как включение в решение некоторой вершины i с не очень хорошим отношением v[i]/t[i] может открыть путь к более быстрому включению других вершин, с большим отношением v/t.

Пусть ГЗ представляет собой цепочечный граф, то есть у каждой вершины такого графа не более одного предка и не более одного потомка. Рассмотрим все возможные цепочки, исходящие из i. Эти цепочки имеют общую начальную часть и различаются тем, сколько вершин в них включено. Каждую такую цепочку можно охарактеризовать суммой вероятностей ее вершин и суммой времен ее вершин. Среди всех цепочек, исходящих из i, найдем ту, которая дает наибольшее отношение суммы вероятностей к сумме времен. Присвоим вершине i две новые характеристики: vsum[i], равную сумме вероятностей цепочки с наибольшим отношением, и tsum[i], равную сумме времен цепочки с наибольшим отношением.

Утверждение 2: При цепочечном ГЗ оптимум можно получить с помощью приоритета – первым в решение включать ту вершину i, отношение vsum[i]/tsum[i] для которой максимально.

Доказательство: пусть на очередном шаге наилучшее отношение

vsum/tsum достигается на вершине i, т.е. vsum[i]/tsum[i]>vsum[k]/tsum[k] для любого k. Сравним качество решений двух решений, полученных:

1) включением на этом шаге вершины k и затем некоторой цепочки, следующей за k, а затем цепочки, начинающейся из i и обеспечившей оценку vsum[i]/tsum[i];

2) включением на этом шаге вершины i, затем цепочки, обеспечившей оценку vsum[i]/tsum[i], затем k и цепочки, следующей за k из предыдущего пункта.

Оценка времени прохождения для пакетов вершин, стоящих в этих двух решениях перед этими двумя цепочками и после них, одинакова. Та часть оценки времени, которая связана с затратами на прохождение пакетами этих вершин предшествующих вершин, одинакова для обоих решений тоже. Оценки времени прохождения для цепочки вершин, следующих за k, увеличатся во втором решении по сравнению с первым на v[k]*tsum[i]+v[k+1]*tsum[i]+...= vsum[k]*tsum[i]. Оценки времени прохождения для цепочки, следующей за i, уменьшатся на vsum[i]*tsum[k]. Так как vsum[i]/tsum[i]>vsum[k]/tsum[k] для любой k и любой цепочки, следующей из k, то второй вариант лучше, и первый можно улучшить, обменяв цепочки, начинающиеся с i и k. Включение двух или более цепочек впереди той, что дала наибольшее vsum[i]/tsum[i], тоже даст худшее решение, так как возможно несколько последовательных улучшений. Следовательно, оптимальная последовательность вершин будет начинаться с вершины i.

При цепочечном графе порядок вершин потомков некоторой вершины i в оптимальном решении был известен, так как однозначно определялся ГЗ. Поэтому с помощью цепочек, построенных из вершин потомков, можно было посчитать приоритет вершины предка. Это не верно для всех графов общего вида. Но для некоторых графов такой порядок удается установить. Пусть ГЗ представляет собой граф, имеющий вид "леса" – то есть у каждой вершины не более одного предка, но может быть более одного потомка.

Рассчитаем приоритет вершин по следующему алгоритму:

Алгоритм 1.

1. Приоритет вершин будет рассчитываться от конца к началу, чтобы ко времени расчета приоритета вершины был известен приоритет всех ее потомков.

2. Построить цепочку из потомков для вершины i по следующим правилам:

2.1. Включить в цепочку саму вершину i.

2.2. Из всех вершин, потомков i (в том числе и транзитивных

потомков), выбрать те, чей предок уже в цепочке. Из полученного множества выбрать ту, у которой наибольшее отношение vsum[k]/tsum[k]. Эту вершину присоединить к цепочке. Так продолжать до тех пор, пока все потомки i не будут упорядочены.

3. Рассчитать приоритет для вершины i с помощью цепочки, полученной в п. 2, используя алгоритм для расчета приоритета цепочечного графа.

Утверждение 3: Если ГЗ имеет форму леса, у некоторой вершины i есть несколько потомков и из каждого потомка далее идет неветвящаяся цепочка, то в оптимальном решении две вершины, которые не упорядочены ГЗ, будут упорядочены в соответствии с vsum/tsum, посчитанными цепочечным алгоритмом.

Доказательство этого утверждения очевидно из цепочечного алгоритма.

Пусть вершина i из утверждения 3 соревнуется за включение с некой другой вершиной j, которая тоже имеет несколько потомков, цепочки за которыми не ветвятся. В оптимальном решении эти вершины будут расположены в соответствии с приоритетами, подсчитанными на цепочках, состоящих из них самих и следующих за ними оптимальных цепочек их потомков. Затем в оптимальном решении будет находиться лучшая вершина из множества, составленного из проигравшей вершины и потомков выигравшей и т.д. То есть порядок вершин из объединения i, j и их потомков в оптимальном решении будет задаваться приоритетами, рассчитанными по следующим за i и j оптимальным цепочкам. Вся эта информация уже известна к моменту расчета общего предка i и j. Поэтому оптимальную последовательность его потомков можно будет рассчитать таким же способом. Так можно продолжать до корня дерева любой высоты и ветвистости. Зная оптимальную последовательность потомков вершины i, можно ввести в граф дополнительные дуги, которые зададут полный порядок на потомках этой вершины. Так как эти дуги не противоречат оптимальному решению, то они не запретят его. Проделаем это со всеми деревьями графа. Затем проведем транзитивную редукцию графа. Удаление транзитивных дуг также не скажется на оптимальном решении. В результате получится цепочечный граф, оптимальное решение которого будет совпадать с оптимальным решением начального графа.

Отсюда следует:

Утверждение 4: Если ГЗ имеет форму леса, то использование приоритетов, описанных в алгоритме 1, дает оптимум.

Временная сложность алгоритма 1 составляет O(n**3).

В полученном алгоритме использовалось свойство, что вершина попадает в множество доступных для включения, после включения ее единственного предка. Это не верно для ГЗ, в котором есть вершины с несколькими предками.

Построить полиномиальный по времени исполнения, точный алгоритм для графов общего вида, т.е. для графа, у которого у вершины может быть несколько предков и несколько потомков, не удалось. (возможно, задача NP полна). Поэтому был разработан алгоритм, использующий метод ветвей и границ.

Для разработки оценок метода ветвей и границ введем понятие “истинного” приоритета вершины на текущем шаге решения. Пусть на текущем шаге в решение будет включена операция i, а затем оставшиеся операции будут включены оптимальным образом, то есть так, чтобы минимизировать цену оставшейся части решения. В таком решении известен порядок всех вершин, стоящих за i. Построим в нем цепочку, начинающуюся из вершины i и продолжающуюся до конца перестановки. В этой цепочке рассмотрим все подцепочки, начинающиеся из i и отличающиеся только длиной. Рассчитаем сумму их вероятностей и сумму их длин. Та цепочка, которая даст максимальное отношение этих сумм (v/t), задаст значение истинного приоритета для i, состоящего из двух величин – vsum[i], tsum[i].

При включении на каждом шаге в решение вершины с наибольшим отношением vsum[i]/tsum[i] получается оптимальное решение.

Для «цепочечных» и «лесных» ГЗ предложенные правила расчета приоритетов дают именно эту величину. Проблема в том, что рассчитать истинный приоритет вершины для графов общего вида не получится, поскольку неизвестна оптимальная перестановка.

Если использовать для определения приоритета вершин приоритет, полученный для «лесных» графов, то возникают следующие проблемы:

1. У некоторых потомков могут быть не включенные в решение предки, не являющиеся потомками вершины, приоритет которой мы рассчитываем. В этом случае такой потомок может оказаться в оптимальной последовательности позже, чем в месте, указанном его “лесным” приоритетом, так как для его включения в решение необходимо сначала включить всех его предков.

2. Существует несколько путей, в ГЗ, из вершины i, для которой ведется расчет приоритета, к некоторому ее потомку. Тогда алгоритм определения последовательности потомков в оптимальном решении не работает, поскольку для его построения явно использовалась древовидная форма подграфа, основанного на потомках.

Отвлечемся от второй задачи и попробуем построить приоритеты для первой. Тут возможны две стратегии поведения. В первой не будем обращать внимание на то, что у некоторых потомков имеются другие предки. В этом случае можно из потомков построить цепочку. На основании этой цепочки рассчитать приоритет. Этот приоритет будет больше или равен истинному приоритету вершины. Назовем его завышенным приоритетом вершины.

По второй стратегии, при достижении потомка, имеющего других предков, он и его потомки просто не включаются в цепочку. Этот прием даст приоритет, меньший или равный истинному приоритету. Назовем его заниженным приоритетом вершины.

По отдельности эти два приема могут служить для построения эвристических алгоритмов. Когда используется один из предложенных приоритетов для выбора вершины, включаемой в решение. Получившееся с помощью эвристического алгоритма решение может быть не оптимальным. Вместе рассчитанный заниженный и завышенный приоритеты можно использовать в методе ветвей и границ. Если у некоторой доступной для включения вершины i завышенный приоритет ниже, чем заниженный у другой доступной для включения вершины j, то ее истинный приоритет меньше, чем истинный у j, и перестановки, в которых i раньше j, можно не рассматривать. Этот приём может сократить вариантность ветвления в методе ветвей и границ, а значит повысить скорость поиска и размерность задач, которые могут быть решены этим методом.

Вторая проблема для графа, имеющего много предков у вершин, может игнорироваться в эвристических методах, поскольку там нужна только эвристическая оценка, и возможны ошибки при расчете приоритета как вверх, так и вниз.

При попытке получить заниженный приоритет, проблема вершины, которая является потомком другой по различным путям в ГЗ, решается просто. Можно не учитывать ее ни в одном из путей. Это может привести только к большему занижению приоритета. Сложней выглядит прием при расчете завышенного приоритета. Пусть идет расчет приоритета некоторой вершины i. В этой работе предлагается допускать включение в формируемую цепочку вершины, у которой есть несколько предков, которые являются потомками i в тот момент, когда первый из этих предков уже включен в цепочку. Это может только еще больше увеличить завышенный приоритет вершины i.

Заметим, что при способе, когда рассчитываются завышенные приоритеты вершин, они не меняются с включением части вершин в частичное решение, поскольку приоритет вершины здесь базируется

только на потомках рассматриваемой вершины, которые не могут быть включены в решение до нее. Поэтому эти оценки можно рассчитать статически, до начала работы метода ветвей и границ. В отличие от завышенных оценок, заниженные оценки можно уточнять в ходе решения, поскольку при включении части вершин в решение некоторые вершины перестают быть вершинами со многими предками и они сами и исходящие из них цепочки могут быть использованы для более точной заниженной оценки. Поэтому для более полного отсеивания вершин из множества доступных для включения на очередном шаге необходимо пересчитывать заниженные оценки на каждом шаге метода ветвей и границ, то есть динамически. К таким алгоритмам предъявляются повышенные требования к быстродействию. Поэтому в данной работе предложен способ, как быстро рассчитать заниженные оценки, возможно несколько ухудшив их в сторону еще большего занижения.

Для этого обратим внимание на тот факт, что оценка вершины i по цепочечному методу может быть заменена гипотезой, что лучшая для i цепочка получается как аргумент максимума vsum/tsum из пары цепочек:

1) лучшей цепочки для ближайшего потомка, к которой добавлена сама вершина i.

2) цепочки, состоящей только из самой вершины i.

Эта гипотеза неверна в общем случае. Ошибка дает погрешность в сторону уменьшения приоритета, так как при расчете максимума просто отбрасывается часть цепочек. Такой приоритет может быть рассчитан для вершины i за время l*l, где l – число непосредственных (не транзитивных) потомков вершины i, не имеющих других предков, кроме i. Квадрат объясняется необходимостью отсортировать потомков i по возрастанию приоритетов, а поскольку их немного, то нет смысла использовать сложные алгоритмы сортировки. Так как l<=n, где n – число вершин графа, то сумма времен, затраченная на расчет приоритета всех вершин sum(l[i]*l[i])<sum(n*l[i])=n*sum(l[i]) <=n*e, где e – число дуг графа, заходящих в вершины, не имеющих несколько предков. У каждой вершины может быть не более одной такой дуги, поэтому e<n, и время работы алгоритма расчета оценок меньше чем n*n.

С целью увеличения скорости работы метода ветвей и границ, кроме способа сокращения множества доступных для включения, в этой работе предложена оценка перспективности продолжения поиска оптимума из текущего частичного решения. Для этого строится полная перестановка вершин, начало которой совпадает с текущим частичным

решением, а оставшиеся вершины добавлены без учета ГЗ, в порядке их приоритетов v[i]/t[i]. Такое упорядочивание может оказаться недопустимым, но даст оценку решения заведомо не худшую, чем любое допустимое решение, которое можно получить из данного частичного решения. Если эта оценка не лучше текущего известного оптимума, то такое частичное решение можно отбросить. Задача расчета оценок будет решаться в динамике, на каждом шаге алгоритма ветвей и границ. Поэтому нужно иметь алгоритм ее быстрого решения. Время, затраченное на расчет такой оценки, состоит из двух частей. Первая часть заключается в расчете оценки для тех вершин, которые вошли в частичное решение. Эта часть времени линейна относительно длины частичного решения. Расчет второй части предполагает необходимость отсортировать в порядке v/t вершины, не включенные в решение. Но этот порядок не зависит от того, какие вершины включены в частичное решение, а какие нет. Поэтому можно создать массив, в котором хранить отсортированные по v/t вершины до начала работы метода ветвей и границ. Теперь при необходимости отсортировать вершины, не включенные в решение, достаточно пройти по этому массиву, пропуская вершины, включенные в частичное решение. Это займет время, линейное по отношению к количеству вершин графа. В результате суммарное время тоже будет линейно по отношению к числу вершин графа.

Были проведены эксперименты как с цепочками, сгенерированными с помощью генератора псевдослучайных чисел, так и с реальными цепочками систем фильтрации. На основе приоритетов, дающих завышенные и заниженные оценки, были запрограммированы эвристические алгоритмы. Их сравнение с алгоритмом, который первой включает в решение вершину, имеющую наилучшее отношение v[i]/t[i] (правило системного администратора), дало 6–8 % выигрыша. Затем был проведен эксперимент, позволивший сравнить эффективность оптимальных решений, полученных методом ветвей и границ, с решениями, полученными с помощью эвристических правил. Эвристические правила проиграли оптимальному решению 0,4 %. Исходные данные модельной задачи не точны. При их вычислении для реальной системы используется предположение, что в текущем периоде частота пакетов, соответствующих различным правилам, сохранится такой же, что и в предыдущем периоде. А это предположение верно только приблизительно. Эта неточность входных повлияет на качество решения сильней, чем потери от использования эвристических алгоритмов вместо поиска оптимума.

Выводы. 1. Разработаны алгоритмы поиска оптимального решения

для частного случая, когда граф ГЗ имеет форму «леса». 2. Разработаны эвристические алгоритмы, позволяющие строить приоритетные правила путем преобразования графов общего вида в графы формы «леса». 3. Разработаны приемы, ускоряющие поиск оптимального решения методом ветвей и границ. 4. Проведен эксперимент, позволяющий оценить эффективность предложенных эвристических алгоритмов как в сравнении с неоптимизированным кодом, так и в сравнении с оптимумом. Показано, что потери качества в сравнении с оптимальным решением при использовании предложенных эвристических алгоритмов незначительно.

Библиографические ссылки

1. Zhenyu Wu, Mengjun Xie, Haining Wang A Fast Dynamic Packet Filter // NSDI ’08: 5th USENIX Symposium on Networked Systems Design and Implementation. San Francisco, California, April 16–18, 2008. San Francisco. 2008. P. 279–292. URL: https://www.usenix.org/legacy/event/nsdi08/tech/full_papers/wu/wu.pdf.

2. Hazem Hamed, Ehab Al-Shaer. Dynamic Rule-ordering Optimization for High-speed Firewall Filtering // ASIACCS '06 Proceedings of the 2006 ACM Symposium on Information, computer and communications security. Taipei. Taiwan. March 21–24. 2006. Taipei. 2006. P. 332–342. URL: [http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.5841&rep=rep1&type=pdf]

3. Шахновский Ю. С., Гончаров А. В. Оптимизация системы фильтрации пакетов // Вестник НТУ ХПИ. 2003. № 6 .Т 1. С. 53–56.

4. Свами М., Тхуласираман К. Графы, сети и алгоритмы. Москва. 1984.

Надійшла до редколегії 23.10.16.

ВІДОМОСТІ ПРО АВТОРІВ

Байбуз Олег Григорович – д-р техн. наук, професор, завідувач кафедри математичного забезпечення ЕОМ Дніпропетровського національного університету імені Олеся Гончара.

Сидорова Марина Геннадіївна – канд. техн. наук, доцент кафедри математичного забезпечення ЕОМ Дніпропетровського національного університету імені Олеся Гончара.

Лапец Ольга Вікторівна – аспірант кафедри математичного забезпечення ЕОМ Дніпропетровського національного університету імені Олеся Гончара.

Долгіх Анастасія Олегівна – аспірант кафедри математичного забезпечення ЕОМ Дніпропетровського національного університету імені Олеся Гончара.

Білобородько Оксана Іванівна – канд. техн. наук, доцент кафедри математичного забезпечення ЕОМ Дніпропетровського національного університету імені Олеся Гончара.

Курочкін Віктор Михайлович – аспірант кафедри прикладної математики навчально-наукового інституту інформаційно-діагностичних систем, Національний авіаційний університет.

Луценко Олег Павлович – канд. техн. наук, асистент кафедри математичного забезпечення ЕОМ Дніпропетровського національного університету імені Олеся Гончара.

Чорна Анастасія Олегівна – магістрант кафедри математичного забезпечення ЕОМ Дніпропетровського національного університету імені Олеся Гончара.

Мацуга Ольга Миколаївна – канд. техн. наук, доцент, доцент кафедри математичного забезпечення ЕОМ Дніпропетровського національного університету імені Олеся Гончара.

Дроздова Ірина Володимирівна – д-р мед. наук, старший науковий співробітник, головний науковий співробітник відділу кардіології державної установи «Український державний науково-дослідний інститут медико-соціальних проблем інвалідності».

Акімова Анна Костянтинівна – магістрант кафедри математичного забезпечення ЕОМ Дніпропетровського національного університету імені Олеся Гончара.

Приставка Пилип Олександрович – д-р техн. наук., професор, завідувач кафедри прикладної математики НДІ ІТТ Національного авіаційного університету.

Чолишкіна Ольга Геннадіївна – канд. техн. наук, доцент кафедри прикладної математики Національного авіаційного університету.

Мартюк Богдан Ігорович – студент кафедри прикладної математики Національного авіаційного університету.

Кузнєцов Павло Володимирович – канд. техн. наук, доцент, доцент кафедри економічної кібернетики та маркетингового менеджменту Національного технічного університету «Харківський політехнічний інститут».

Шахновський Юрій Сергійович – канд. техн. наук, доцент, доцент кафедри системного аналізу та управління Національного технічного університету «Харківський політехнічний інститут».

ЗМІСТ

Наукове видання

АКТУАЛЬНІ ПРОБЛЕМИ АВТОМАТИЗАЦІЇ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ТОМ 20

Збірник наукових праць

Українською, російською та англійською мовами

Підписано до друку 14.12.2016. Формат 60x84/16. Друк цифровий.
Ум. друк. арк. 6,16.Тираж 100 прим. Зам. № 277

Видавництво та друкарня "Ліра" .
49000, м. Дніпропетровськ, вул. Наукова, 5
Свідоцтво Державної реєстрації ДК 188 від 19.09.2000