

МЕТОД І АЛГОРИТМИ ПЕРЕТВОРЕННЯ ТИПУ "КОНТУР–КОНТУР" ТА "ОБЛАСТЬ–ОБЛАСТЬ"

© Батько Ю., 2011

Запропоновано метод і алгоритми перетворення контурів та областей. Наведено результати перетворення контурів та областей на біомедичних зображеннях.

Ключові слова: контур, область, скелет об'єкта, перетворення, алгоритми, мінімальний скелет, ізоморфні скелети.

The method and algorithms contours and areas transformation are offered in the article. The results of contours and areas transformation are described.

Keywords: contour, area, skeleton, transformation, algorithms, isomorphic skeletons.

Вступ

Швидкі темпи розвитку функціональних можливостей та зниження вартості засобів обробки цифрової інформації забезпечило широке впровадження цифрових технологій у різні галузі людської діяльності. Використання цифрових технологій дало змогу зменшити вплив людського фактора в технологічних процесах, проте породило ряд нових задач. Однією з важливих задач на рівні з виділенням, розпізнаванням та синтезом об'єктів на цифрових зображеннях є задача побудови достовірного опису об'єкта. Підходи до побудови опису об'єктів представлені в роботах багатьох вітчизняних та закордонних вчених: У. Претта [1], Р. Гонсалеса [2], В. Сойфера [3], Д. Форсайта [4], Я. Фурмана [5], К. Фу [6] та інших [7–9]. Відомі декілька підходів опису об'єктів: на основі характеристичних ознак (метричні, топологічні тощо) [1–5], структурного опису [6] тощо. У задачах, для яких необхідні дані про форму, використовується опис на основі скелету та контуру об'єкта [10–13].

Опис об'єкта за допомогою скелета та контуру є одним з ефективних підходів для аналізу зображень. За допомогою аналізу контуру можна визначити різні метричні характеристики об'єкта: периметр, площу, колоподібність тощо. Аналіз скелета дає змогу виявити кількість зв'язних компонент, розташування, направленість об'єкта.

Інформація про скелети та контури широко використовується в задачах розпізнавання текстових даних, автоматичного аналізу топологічних карт, медицині тощо. Окремою задачею, де використовуються скелети та контури, є перетворення об'єктів [14–15].

Перетворення об'єктів на основі інформації про контур обмежені афінними перетвореннями: масштабуванням, переносом, поворотом, відображенням. Проте апарату афінних перетворень не завжди достатньо для виконання складних перетворень, наприклад, зменшення (збільшення) окремих фрагментів об'єкта [15]. Тому для складних перетворень доцільніше використовувати скелети, оскільки вони дають змогу перетворювати окремі гілки скелета, що відповідають за різні фрагменти об'єкта, не порушуючи при цьому його цілісності. Похибка перетворення при цьому є мінімальною.

Метою роботи є розроблення методу та алгоритмів перетворення типу "контур–контур" та "область–область".

Метод та алгоритм перетворення типу „контур–контур”

Для знаходження функції перетворення контуру $C_i \xrightarrow{Q_{ij}} C_j$ розіб'ємо початкові контури на сегменти. Отримаємо $C_i = \bigcup_{r=1}^m D_{ir}$ та $C_j = \bigcup_{l=1}^n D_{jl}$, $i, j, m, n \in N$, причому необхідною є умова рівності

кількості сегментів $m = n$ (рис. 1). Кожний сегмент D_{ir} та D_{jl} заданий у вигляді полінома степеня $k \leq 3$. Для кожної пари відповідних сегментів функція перетворення матиме наступний вигляд $D_{j1} = Q_{ij1}(D_{i1}), D_{j2} = Q_{ij2}(D_{i2}), \dots, D_{jn} = Q_{ijn}(D_{in})$ (рис. 1).

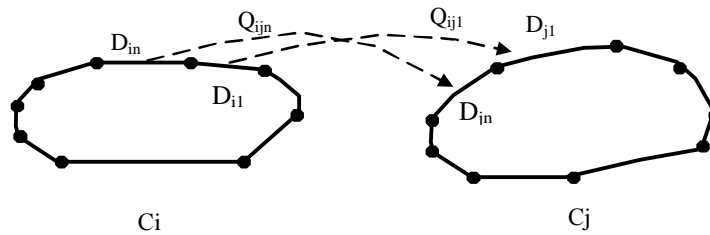


Рис. 1. Розбиття контурів об'єктів на сегменти

Виділимо на контурі C_i сегмент D_{i1} на проміжку $[a_1; b_1]$, який представлений функцією $y = j_{i1}^k(x)$ відповідно на контурі C_j сегмент D_{j1} на проміжку $[a_2; b_2]$, який представлений функцією $y = j_{j1}^k(x')$ (рис. 2).

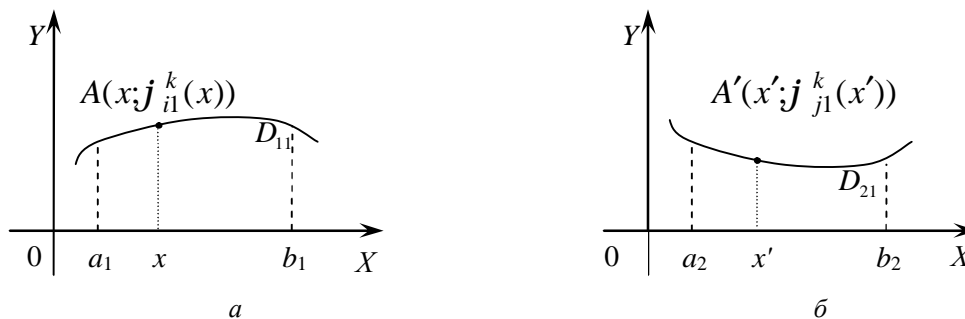


Рис. 2. Приклад виділення сегментів на контурі

Тоді перетворення $Q_{ij}(x, y)$ дорівнює $Q(x, y) = (x', y')$. Абсциса x' дорівнює $x' = \frac{b_2 - a_2}{b_1 - a_1}(x - a_1) + a_2$, ордината $j_{j1}^k(x')$, тобто

$$Q_{ij}(x, y) = \left(\frac{b_2 - a_2}{b_1 - a_1}(x - a_1) + a_2; j_{j1}^k(x') \right).$$

Для отримання функцій перетворення типу „контур – контур” необхідно виконати такі кроки:

1. Розбиваємо вхідне зображення $I = \{(x, y) | 1 \leq x \leq m, 1 \leq y \leq k\}$, де m і k – висота і ширина вхідного зображення, на область фону та об'єкти.

2. Визначаємо контури виділених об'єктів шляхом проходження контурів. У результаті отримуємо масив контурів $\{C_i\}$ та внутрішніх областей об'єктів $\{O_i\}$.

3. Розбиваємо зовнішні контури $\{C_i\}$ на сегменти $\{C_i\} = \{\bigcup_{r=1}^m D_{ir}\}$. Кількість сегментів на

контурах повинна бути однаковою. Якщо кількість сегментів двох контурів є різною, то додатково розбиваємо контур з меншою кількістю сегментів. Для додаткового поділу можна використати сегменти, які мають максимальне значення середньої кривизни.

4. Знаходимо функцію перетворення $Q_{ij,n}$ для кожної з пар сегментів D_{im} та D_{ik} , $m, k = 1..n$ контурів C_i та C_j відповідних об'єктів, де n – кількість сегментів пари контурів, на основі формули:

$$Q_{ij} = \frac{x_2 - x_1}{x_2 - x_1} * x - \frac{x_2 - x_1}{x_2 - x_1} * x_2 + x_2$$

Метод перетворення типу „область–область”.

Нехай дано два об’єкти Obj_1 та Obj_2 (рис.4). Згідно з теоремою Жордана для замкнутих областей існує перетворення, яке перетворює одну область на іншу, одну підобласть на іншу під-область $O^* = T(O)$ [15].

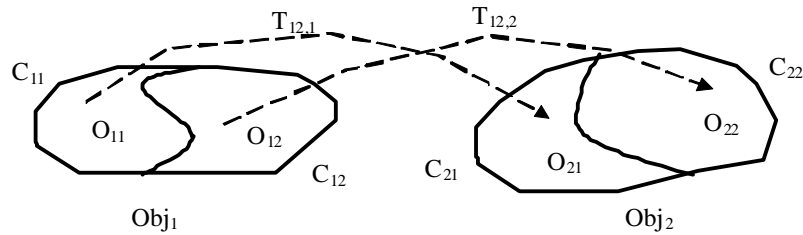


Рис. 3. Перетворення типу “ область–область”

Розіб’ємо вхідні об’єкти на однорідні області $Obj_1 = O_{11} \cup O_{12} \cup \dots \cup O_{1n}$ та $Obj_2 = O_{21} \cup O_{22} \cup \dots \cup O_{2m}$, n та m – кількість однорідних областей, що формують відповідний об’єкт. Для існування перетворення необхідна однакова кількість областей та еквівалентність матриць інцидентій областей. Тоді отримуємо $O_{21} = T_{12,1}(O_{11}), O_{22} = T_{12,2}(O_{12}), \dots, O_{2m} = T_{12,n}(O_{1n})$, де $T_{ij,k}$ – перетворення k -ї однорідної області об’єкта Obj_i до k -ї однорідної області об’єкта Obj_j . Як показали експериментальні дослідження, в об’єктах зі складною контурною лінією (велика кількість випуклостей) використання афінних перетворень приводить до значної похибки перетворення [16]. Отже, для таких об’єктів доцільніше використовувати топологічний простір та замість областей аналізувати їх скелети, що дають змогу зменшити похибку перетворення об’єктів. Введемо l_∞ -метрику, для якої справедлива формула: $|A - B| = \max\{|x_1 - x_2|, |y_1 - y_2|\}$, де (x_1, y_1) – координати точки A , (x_2, y_2) – координати точки B . Отже, точка належить скелету області, якщо вона є центром максимального вписаного в область квадрата із сторонами, паралельними координатним осям.

У роботі [17] показано, що для кусково-лінійних апроксимованих контурів C_1 і C_2 скелети областей O_1 і O_2 є кусково-лінійними. Крім цього, скелети $S_1(O_1)$ і $S_2(O_2)$ є графами без циклів (деревами).

Отже, скелети $S_1(V_1, E_1)$ та $S_2(V_2, E_2)$ є деревами, де V_1, V_2 – множини вершин, а E_1, E_2 – множини ребер (гілок).

Однією з основних умов проведення перетворення типу „область–область” є ізоморфність скелетів.

Скелети $S_1(V_1, E_1)$ та $S_2(V_2, E_2)$ називаються **ізоморфними**, якщо існує таке взаємно однозначне відображення j множини вершин V_1 на множину вершин V_2 , що ребро $(v, w) \in E_1$ тоді і тільки тоді, коли ребро $(j(v), j(w)) \in E_2$.

У загальному випадку два скелети $S_1(V_1, E_1)$ та $S_2(V_2, E_2)$ не є ізоморфними.

Узагальнений алгоритм перетворення типу „область–область”.

1. Виділяємо на зображенні об’єкти $\{Obj_j\}$, визначаємо їх однорідні області $\{O_{ij}\}$ та контури $\{C_{ij}\}$.
2. Проводимо кусково-лінійну апроксимацію контурів однорідних областей.
3. Знаходимо скелети однорідних областей $S_{ij}(O_{ij})$.
4. Якщо скелети відповідних областей не є ізоморфні, то приводимо їх до ізоморфного вигляду.

5. Знаходимо функції перетворення для відповідних гілок скелетів. Функції перетворення обчислюються за формулою $T = \frac{x_2 - x_1}{x_2 - x_1} * x - \frac{x_2 - x_1}{x_2 - x_1} * x_2 + x_2$.

6. Проводимо перетворення для будь-якої точки D області O_1 у відповідну точку D' області O_2 .

Перетворення будь-якої точки D області O_1 у відповідну точку D' області O_2 проводиться шляхом знаходження проекції точки D на відповідний сегмент контура (точка A) та гілки скелета (точка B) [16]. Координати точки D обчислюються через координати точок A і B :

$D = tA + (1-t)B$, де $t = \frac{|D-B|}{|A-B|}$. Координати точок A' і B' області O_1 дорівнюють: $A' = T_1(A)$, $B' = T_2(B)$. Відповідно координати точки D' дорівнюють: $D' = tT_1(A) + (1-t)T_2(B)$.

Перетворення T_{12} області O_1 на область O_2 для будь-яких точок $D \in O_1$ і $D' \in O_2$ дорівнює:

$$T_{12} = \frac{tT_1(A) + (1-t)T_2(B)}{tA + (1-t)B}$$

Алгоритм скелетизації

1. Контур C_i апроксимуємо за допомогою кусково-лінійної апроксимації:

$$C_i \xrightarrow{\text{апроксимація}} C_i'$$

2. Послідовно проходимо кожною точкою, що належить апроксимованому контуру $C_i'(x, y), x, y \in N$, відкладаючи відрізки, паралельні осям OX та OY . Кінцем відрізка вважається остання точка, яка належить відріжку та контуру об'єкта. На основі відкладених відрізків пробуємо вписати квадрати з максимальною площею. Точка перетину діагоналей вписаного квадрата належатиме скелету S_i об'єкта (рис.4).

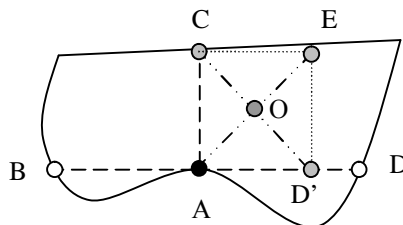


Рис. 4. Приклад визначення точок, що належать скелету

3. За допомогою морфологічних операцій нарощування та утоншення отримуємо одно-зв'язний скелет товщиною в одну точку.

4. Проводимо відсікання малоінформативних гілок на основі критеріїв втрати площі (1) та критерію втрати контурної лінії (периметра) (2).

$$K_A = \left(1 - \frac{A_i}{A_o}\right) * 100\% , \quad (1)$$

де A_i – площа об'єкта, відновленого на основі скелета, без i -ї кінцевої гілки; A_o – площа об'єкта, відновленого на основі скелета.

$$K_P = \left(1 - \frac{P_i}{P_o}\right) * 100\% , \quad (2)$$

де P_i – довжина контуру об'єкта відновленого на основі скелета, без i -ї кінцевої гілки; P_o – довжина контуру об'єкта, відновленого на основі скелета.

Результатом роботи алгоритму скелетизації є однозв'язний мінімальний скелет об'єкта товщиною в одну точку.

Отриманий скелет, як правило, неоднорідний, тому проводиться процедура з'єднання окремих елементів скелету. Для утворення однорідного скелета необхідно з'єднати кінцеві вузли окремих елементів скелета. Кінцевим вузлом вважатимемо вузол, що належить скелету об'єкта та має тільки одного сусіда, що також належить скелету. Якщо вузол має менше ніж вісім сусідів (лежить на краю зображення), то відсутні сусідні точки визнаються такими, що не належать скелету об'єкта.

Алгоритм знаходження мінімального скелета. Мінімальним скелетом вважатимемо скелет з мінімальною кількістю ребер, що описує об'єкт з допустимою похибкою.

Для визначення мінімального скелета необхідно виконати такі кроки:

1. Визначаємо множину кінцевих ребер скелета $\{E_{end i}\}$. Кінцевим ребром вважатимемо ребро, яке є інцидентним до кінцевого вузла скелета.

2. Для кожного кінцевого ребра визначаємо коефіцієнти втрати площі K_A (1) та втрати контурної лінії (периметра) K_P (2).

3. Визначаємо кінцеве ребро, для якого коефіцієнти втрати площі та контурної лінії будуть мінімальними $K_A(E_{end i}) \rightarrow \min$ або $K_P(E_{end i}) \rightarrow \min$.

4. Якщо коефіцієнт втрати площі менший за допустиме порогове значення $K_A(E_{end i}) \leq \Delta A$ та коефіцієнт втрати контурної лінії менший за допустиме порогове значення $K_P(E_{end i}) \leq \Delta P$, то це ребро видаляється зі скелета як малоінформативне. Повернення на крок 2.

5. Якщо коефіцієнт втрати площі більший за допустиме порогове значення $K_A(E_{end i}) > \Delta A$ або коефіцієнт втрати контурної лінії більший за допустиме порогове значення $K_P(E_{end i}) > \Delta P$, то алгоритм завершує свою роботу.

Результатом роботи алгоритму є мінімальний скелет об'єкта. На рис. 5 зображено роботу запропонованого алгоритму відсікання малоінформативних ребер скелета.

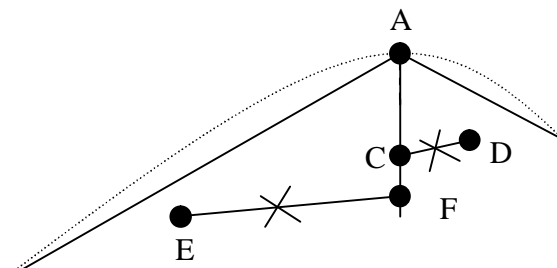


Рис. 5. Приклад алгоритму відсікання малоінформативних гілок

Алгоритм приведення неізоморфних скелетів до ізоморфного вигляду. Для проведення перетворень виду "скелет–скелет" необхідною умовою є ізоморфність двох скелетів $S_1(V_1, E_1)$ та $S_2(V_2, E_2)$. У загальному випадку пари скелетів можуть мати такий вигляд:

скелети $S_1(V_1, E_1)$ та $S_2(V_2, E_2)$ мають однакову кількість вузлових вершин $V_1 = V_2$ та однакову кількість гілок $E_1 = E_2$;

– скелети $S_1(V_1, E_1)$ та $S_2(V_2, E_2)$ мають різну кількість вузлових вершин $V_1 \neq V_2$ та однакову кількість гілок $E_1 = E_2$;

– скелети $S_1(V_1, E_1)$ та $S_2(V_2, E_2)$ мають однакову кількість вузлових вершин $V_1 = V_2$ та різну кількість гілок $E_1 \neq E_2$;

– скелети $S_1(V_1, E_1)$ та $S_2(V_2, E_2)$ мають різну кількість вузлових вершин $V_1 \neq V_2$ та різну кількість гілок $E_1 \neq E_2$;

Для приведення скелетів до ізоморфного виду використовується такий алгоритм:

1. Для скелетів $S_1(V_1, E_1)$ та $S_2(V_2, E_2)$ знаходимо вузлову точку з максимальним степенями $V_1 \max$ та $V_2 \max$. Якщо в скелеті присутні дві вузлові вершини з однаковими степенями, то максимальною вважають вершину з більшою вагою $g(V_i)$.

Вага вузлової точки $g(V_i)$ визначається на основі ваг всіх інцидентних до неї гілок (наприклад як сума всіх їх ваг). Вагу гілки $g(E_i)$ обчислюємо на основі ваг точок, що формують її $e_j \in E_i$. Для визначення ваги, наприклад, можна використати середньоарифметичне значення ваг точок, які належать гілці.

2. Представляємо скелети $S_1(V_1, E_1)$ та $S_2(V_2, E_2)$ у вигляді дерев. Коренями відповідних дерев є точки $V_1 \max$ та $V_2 \max$. Розбиваємо отримані дерева на рівні l_i (рис.6).

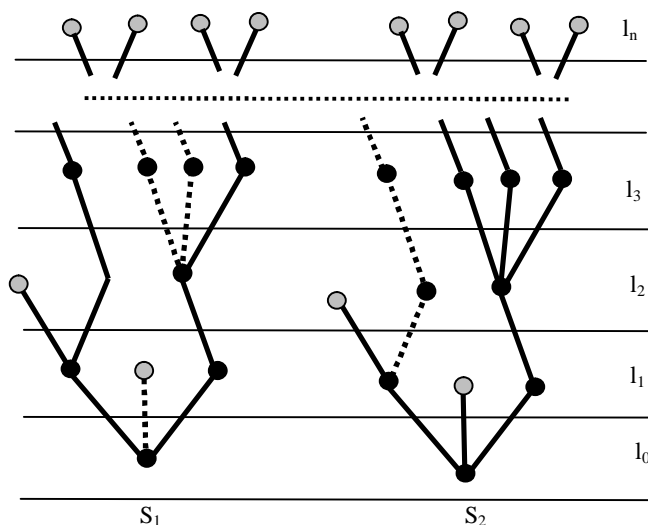


Рис. 6. Представлення скелетів у вигляді дерев.
Пунктирні лінії – штучно додані гілки

3. Сортуємо гілки на нульовому рівні. Гілки сортують за зростанням степеня відповідної вузлової точки зліва направо. Якщо степені двох або більше вузлових точок однакові, то такі гілки сортуються на основі значення ваги відповідної вузлової точки.

4. Приводимо до ізоморфного вигляду l_i рівня дерева. Цю процедуру здійснюється шляхом додавання відсутніх гілок та вузлових точок до відповідного дерева. Оскільки дерева відображають мінімальні скелети, то використання операції ВИДАЛЕННЯ (гілка, вузлова точка) не можливе.

5. Операцію ДОДАВАННЯ (гілка, вузлова точка) виконуємо за рівнями, що передбачає додавання відсутніх гілок та вузлових точок скелета. Ваги доданих гілок встановлюються такими, що незначно впливають на похибки нарощування площі мікрооб'єкта $g(V_i) \rightarrow \min$.

6. Вузлова точка, до якої необхідно додати штучно проведену гілку, визначаємо на основі двох параметрів: степеня вузла та ваги гілки $g(V_i)$.

Оскільки необхідною і достатньою умовою ізоморфності скелетів є однакова кількість вершин з однаковими степенями на відповідних рівнях, то результатом цього алгоритму є два ізоморфні скелети $S_1(V_1, E_1)$ та $S_2(V_2, E_2)$.

Приклад роботи запропонованих алгоритмів

Проаналізуємо результати роботи алгоритмів на прикладі цитологічних зображень (рис.7). Серед особливостей цього типу зображень є наявність подібних об'єктів (клітин), що дає змогу використовувати їх для досліджень запропонованих алгоритмів.

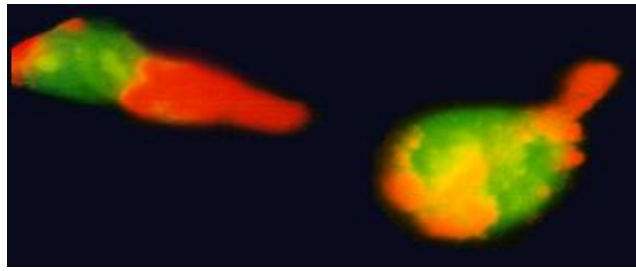
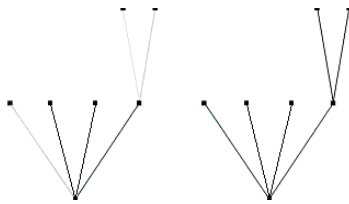


Рис. 7. Приклад вхідного зображення

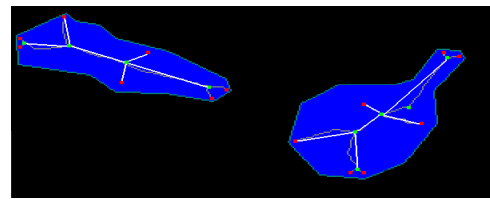
Виділяють скелет на основі вписування квадратів у межі об'єктів. При цьому виконується умова, що хоча б дві вершини квадрата лежать на контурі об'єкта. Массив пікселів, що відповідають точкам перетину діагоналей квадратів, утворює скелет об'єкта.

Для видалення надлишкових та малоінформативних гілок проводимо процедуру відсікання гілок. Гілка вважається малоінформативною, якщо відсоток втраченої об'єктом площі при відновленні на основі скелета буде меншим за деяке порогове значення Δ .

Для визначення коефіцієнтів перетворення необхідно привести скелети об'єктів до ізоморфного вигляду. Для цього представляємо скелети у вигляді дерев. Коренем дерева вважаємо вершину з максимальним степенем. Деревя зводяться до ізоморфного вигляду шляхом додавання штучних гілок. Приклад приведення до ізоморфного вигляду наведено на рис. 8. На рис. 8, а гілки, позначені чорним кольором, відповідають початковому дереву, а сірим – штучним. Вага штучних гілок встановлюється близькою до 0.



а – приклад нарощування двох не ізоморфних дерев.
Чорні лінії – основні лінії, сірі – додані штучно



б – триклад додавання штучних гілок до скелета

Рис. 8. Приведення до ізоморфного вигляду двох неізоморфних скелетів

Приведення скелетів до ізоморфного стану (рис. 8, б) дає змогу визначити пари відповідних гілок скелетів, що є необхідною умовою для знаходження коефіцієнтів перетворення типу „область–область”.

Висновки

Запропоновані методи та алгоритми дають змогу здійснювати перетворення типу „контур–контур” та “область–область” для областей із складними кусково-лінійними апроксимованими контурами. Розроблені алгоритми виділення мінімального скелета на основі критеріїв втрати площі та периметра дали змогу зменшити похибку виділення на 10% порівняно з відомими аналогами. Похибка приведення двох неізоморфних скелетів до ізоморфного вигляду становить менше 4%.

1. Pratt W.K. *Digital image processing: PIKS Scientific inside, 4th ed* / William K. Pratt. – USA: John Wiley & Sons – 2007, – 782 p. 2. Форсайт Д. *Компьютерное зрение. Современный поход* / Дэвид А. Форсайт, Джин Понс. – М.: Вильямс. – 2004. – 928 p. 3. Фу К. *Структурные методы в распознавании образов* / К. Фу. – М.: Мир, 1977 – 320с. 4. Шатиро Дж. *Компьютерное зрение* / Дж. Шатиро, Дж. Стокман. – М.: Бином. Лаборатория знаний, 2006. – 752с. 5. Яне Б. *Цифровая обработка изображений* / Б. Яне. – М.: Техносфера, 2007. – 584 с. 6. Глушков Г.А. *Компьютерная графика* / Глушков Г.А. – Харьков: Фолио, 2000. – 500с. 7. Мищенко А.С. *Краткий курс дифференциальной геометрии и топологии.* / А.С. Мищенко, А.Т. Фоменко. – М.: Физматлит, 2004.

– 304 с. 8. Свами М. Графы, сети и алгоритмы: Перевод с английского. // М. Свами, К. Тхуласираман – М.: Мир, 1984 – 455 с. 9. Местецкий Л. Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры // Л. Местецкий. – М.: ФИЗМАТЛИТ, 2009 – 288 с. 10. Blum H. A Transformation for Extracting New Descriptors of Shape, Models for the perception of Speech and Visual Form // H. Blum. – MIT Press. – 1967. – P. 363–380. 11. Harrison S.J. The influence of shape and skeletal axis structure on texture perception. // Sarah J. Harrison – Journal of Vision – 2009 – 9 – P.1–21. 12. Rizvandi N. B. Skeleton analysis of population images for detection of isolated and overlapped nematode celegans. / N. B. Rizvandi, A. Pizurica, F. Rooms, W. Philips // 16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland – August – 2008 pp.25-29. 13. Žitkevičius E. On skeletonization of blood vessels in angiographic mri images of human brain // E. Žitkevičius, D. Grigaitis, D. Navakauskas – Information technology and control, 2007 – pp. 372-376. 14. Bai X. Skeleton Pruning by Contour Partitioning with Discrete Curve Evolution // X. Bai, L. J. Latecki, W.-Yu. Liu / IEEE transactions on pattern analysis and machine intelligence, vol. 29, – no. 3, - march – 2007 – P.1–14. 15. Местецкий Л.М. Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры // Л.М. Местецкий. – М.: ФИЗМАТЛИТ, 2009. – 288 с. 16. Березький О.М. Порівняння алгоритмів перетворення зображень в афінному та топологічному просторах // О.М. Березький, Ю.М. Батько. – “Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта (ISDMCI’2011). – Євпаторія, 2011. – С. 68–72. 17. Березский О.Н. Топологические методы и алгоритмы преобразования контуров и областей плоских изображений / О.Н. Березский // Проблемы информатики и управления. – 2010. – №5. – С.123–131.

УДК 681.14

В. Коцовський

Ужгородський національний університет

АЛГОРИТМІЧНА СКЛАДНІСТЬ ЗАДАЧІ НАВЧАННЯ ДВОПОРОГОВИХ НЕЙРОНІВ

© Коцовський В., 2011

Розглядаються питання, пов’язані з розпізнаванням скінченних множин за допомогою двопорогових нейронних елементів. Показано, що задача навчання ДНЕ є NP-повною. Також наведено умови, виконання яких забезпечує двопороговість булевих функцій, які задаються за допомогою списків рішень.

Ключові слова: двопороговий нейронний елемент, порогова логіка, нейронна мережа, теорія складності алгоритмів, список рішень.

We study finite set dichotomies on bithreshold neurons. We prove that training a BN is NP-complete task. We also give sufficient conditions ensuring that decision list represents a bithreshold function.

Key words: bithreshold neuron, threshold logic, neural networks, complexity theory, decision list.

Вступ і формулювання задачі

Нейромережі, побудовані з нейронних елементів (НЕ), успішно використовуються для розв’язування широкого кола практичних задач [1]. При цьому залежно від специфіки задачі вибирається та чи інша активаційна функція НЕ. Історично першою було використано порогову функцію активації, прототипом якої були процеси активації нервових клітин живих організмів. На її основі Розенблат [2] побудував модель розпізнавального пристрою, який отримав назву перцептрон. Було запропоновано алгоритм навчання цього пристрою, – так званий алгоритм навчання з підкріпленням, який з успіхом використовувався для розв’язування задач розпізнавання. Збіжність цього алгоритму було встановлено у роботі [3]. Проте незабаром Мінський [4] показав,