

КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

V.P. Boyun, P.Y. Sabelnikov,
Y.A. Sabelnikov

APPROXIMATION OF CLOSED CONTOUR IMAGE BY POLYGON

Method for approximating a closed contour image of a polygon, whose vertices are directly this point of the image are propose.

Key words: Approximation, contour, polygon.

Предложен метод аппроксимации замкнутого контурного изображения многоугольником, вершинами которого являются непосредственно точки этого изображения.

Ключевые слова: аппроксимация, контур, многоугольник.

Запропоновано метод аппроксимції замкненого контурного зображення багатокутником, вершинами якого є безпосередньо точки цього зображення.

Ключові слова: аппроксимация, контур, багатокутник.

© В.П. Боюн, П.Ю. Сабельніков,
Ю.А. Сабельніков, 2013

УДК 004.932

В.П. БОЮН, П.Ю. САБЕЛЬНИКОВ, Ю.А. САБЕЛЬНИКОВ

АПРОКСИМАЦІЯ ЗАМКНЕНОГО КОНТУРНОГО ЗОБРАЖЕННЯ БАГАТОКУТНИКОМ

Вступ. У роботі пропонується метод апроксимації замкненого контурного зображення багатокутником, вершинами якого є безпосередньо точки цього зображення. Критерій апроксимації формулюється наступним чином: відстань від кожної точки контуру до найближчого апроксимуючого відрізка не повинна перевищувати похибку апроксимації ϵ . При цьому довжина апроксимуючих відрізків має бути як можна більшою, а кількість вузлів апроксимації найменшою.

У літературі представлені різні методи апроксимації контурів. Розглянемо два з них, які застосовані на вище вказаному критерії апроксимації і відповідають поставленій задачі.

У роботі [1] представлено модифікований метод найменших квадратів для виділення прямих ліній на контурних зображеннях, який полягає в наступному.

Нехай є дві точки P_1 і P_2 . За цими точками будується рівняння прямої і знаходяться коефіцієнти A і b , а також тип рівняння ($y = Ax + b$) або ($x = Ay + b$). Для кожної послідууючої точки P_i перевіряється відстань від неї до цієї прямої. Якщо вона менше деякого порога ϵ , то ця точка приєднується до множини точок прямої без перерахунку її коефіцієнтів. У протилежному випадку перебуває рівняння прямої з урахуванням останньої точки і перевіряється відстань всіх попередніх точок до нової прямої. Якщо хоч одна з них більше ϵ , то остання точка є хибною, не приєднується до множини точок прямої і процес завершується.

Недоліком цього методу є те, що іноді перевіряються на відстань від прямої всі попередні точки, тому кількість обчислень буде пропорційна n^2 , де n – кількість точок відрізка контуру, що апроксимується. Запропонована для нього умова закінчення процесу апроксимації також не ефективна, тому що при подальшому аналізі може знайтися більш віддалена кінцева точка апроксимуючого відрізка, що відповідає критерію не перевищення похибки апроксимації. Тому здобувається більша кількість вузлів апроксимації.

Найбільш близьким до запропонованого нами методу є метод розглянутий в [2]. Але в ньому розглядається алгоритм, де як похибка апроксимації береться відстань, яка дорівнює шагу “піксельної” сітки. Якщо до цього методу застосувати довільну похибку ε , то потрібна додаткова умова, щоб коректно прийняти рішення про кінцеву точку апроксимуючого відрізка, що відбивається на практичній реалізації алгоритму.

Метод, що пропонується, заснований на раніше вказаному критерії апроксимації і послідовному аналізі точок замкненого контуру, при цьому обчислення проводяться тільки для чергової точки не повертаючись для перевірки критерію до всіх попередніх точок.

Суть методу полягає у послідовному пошуку можливих напрямків і кінцевих точок апроксимуючого відрізка прямої лінії, що виходить з чергової початкової точки A_z (рис. 1).

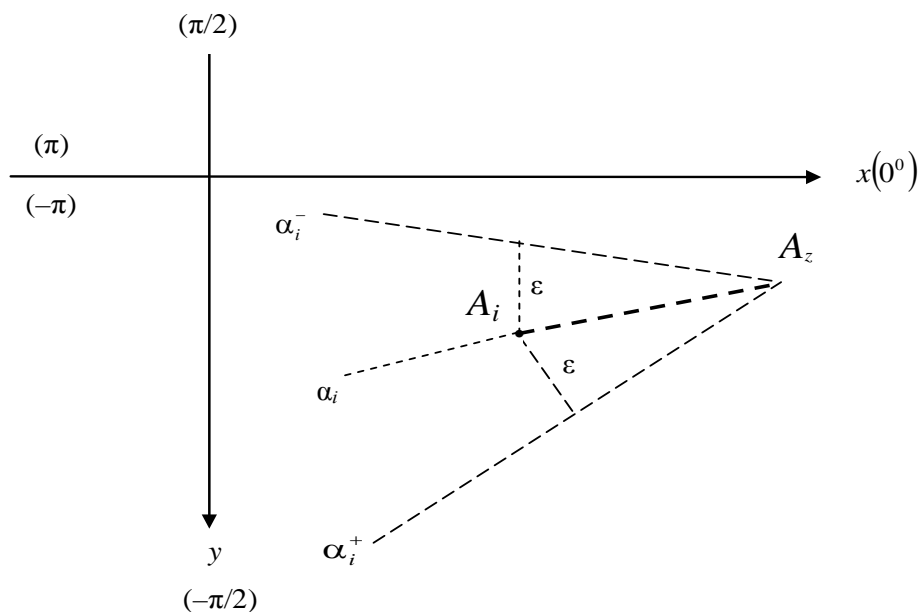


РИС. 1

З цією метою для кожної точки A_i формується радіальний растр, тобто направлення двох крайніх променів, що виходять із A_z і знаходяться на відстані похибки ε від точки A_i . Всі промені в межах цього растру знаходяться від точки A_i на відстані не більшій ніж ε . Також формується черговий растр, як пересічення растрів цієї і всіх попередніх точок контуру, починаючи з точки, що лежить від A_z на відстані більше ніж ε . Всі промені цього растра є можливими напрямками апроксимуючої прямої. Існування чергового растра є умовою існування променів, від яких всі попередні точки знаходяться на відстані не більше ε . Якщо на черговій точці растр зійшовся, тобто $(\alpha_r^+) < (\alpha_r^-)$, то остання точка A_k і є кінцевою точкою апроксимуючого відрізка і початковою нового.

Точка A_i – це можлива кінцева точка апроксимуючого відрізка (тобто $k: = i$) за наступними умовами:

- точка A_i лежить у межах попереднього чергового растра;
- відстань від точки A_z до A_i більша або дорівнює максимальній відстані від A_z до всіх попередніх точок.

При виконанні цих умов можна стверджувати, що всі точки контуру між A_z і A_k знаходяться від відрізка прямої, що з'єднує ці точки, на відстані не більшій ε .

Другу умову можна вибрати іншою, що не суперечить критерію апроксимації, але з практичних зображень автори вибрали саме цю. Вона дозволяє менш точно відслідковувати тупі кути і більш точно гострі, запобігаючи їх зрізанню.

Більш детально розглянемо метод на прикладі алгоритму, розподіливши його на окремі кроки. Введемо позначення:

- n – кількість точок контуру;
- i – індекс точки контуру;
- k – індекс можливої кінцевої точки чергового апроксимуючого відрізка;
- z – індекс початкової точки чергового апроксимуючого відрізка;
- $l(A_z, A_i)$ – відстань між точками A_z та A_i ;
- l_{max} – максимальна відстань від точки A_z до точок, що проаналізовані перед точкою A_i ;

ε – похибка апроксимації;

α_i – кут променя, що з'єднує точки A_z і A_i ;

$\{\alpha_i^+, \alpha_i^-\}$ – растр для точки A_i (α_i^+, α_i^- – кути крайніх променів растра);

$\{\alpha_r^+, \alpha_r^-\}$ – черговий растр (α_r^+, α_r^- – кути крайніх променів растра).

Для коректності зауважимо наступне:

- всі кути беруться відносно координатної сітки, представленої на рис. 1;
- аналізуються тільки ті точки A_i , які лежать від A_z на відстані більшій ніж ε , тому що точки, які лежать від A_z на відстані $\leq \varepsilon$, також знаходяться на відстані $\leq \varepsilon$ від апроксимуючого відрізка прямої;

- всі операції додавання і віднімання кутів мають бути груповими, тобто якщо кути лежать у діапазоні від $-\pi$ до $+\pi$, то і результати операцій мають бути в цьому діапазоні;

- оскільки відстань між A_z і A_i більша ніж ε , то $\left|(\alpha_i^+) - (\alpha_i^-)\right| < \pi$.

Опис алгоритму апроксимації за кроками.

1. Початковій точці контуру присвоюється індекс 0. Найбільш для цього підходить точка з максимальною відстанню від центру площі замкненої контуру, тому що у випадку багатогранних фігур вона практично завжди є кутковою. Її беруть як початкову точку чергового апроксимуючого відрізка A_z ($i := 0, z := 0$).

2. Координати A_z записуються у вектор апроксимації. Починаючи від A_z йдуть послідовно по контуру (в даному випадку проти годинної стрілки) і шукають точку, яка знаходиться на відстані від початкової більше ніж ε . Звісно, що початкова точка належить апроксимуючому відрізку і для всіх точок, які лежать від неї на відстані не більше ε , критерій апроксимації задовольняється. Якщо така точка не знайдена і всі точки контуру проаналізовані, здійснюють вихід з процедури.

Для знайденої точки A_i обчислюється растр $\{\alpha_i^+, \alpha_i^-\}$, тобто напрямлення двох крайніх променів, які виходять із A_z і знаходяться на відстані від знайденої точки не більше ніж ε (дивись рис. 1).

Значення цього растра надається черговому растру $\{\alpha_r^+, \alpha_r^-\}$, фіксується індекс знайденої точки ($k := i$), як індекс можливої кінцевої точки апроксимуючого відрізка і змінній l_{max} присвоюється значення відстані $l(A_z, A_i)$.

3. Береться наступна точка і перевіряється її відстань від A_z . Якщо ця відстань $l(A_z, A_i) \leq \varepsilon$, переходять до наступної точки. Якщо ні, то перевіряється умова $l_{max} \leq l(A_z, A_i)$. При виконанні цієї умови виконуються операції:

- $l_{max} := l(A_z, A_i)$;
- обчислюється кут α_i ;
- якщо $(\alpha_r^-) \leq \alpha_i \leq (\alpha_r^+)$, то $k := i$, у протилежному випадку значення k не змінюється.

При $l_{max} > l(A_z, A_i)$ перелічені операції не виконуються. Обчислюється для взятої точки растр $\{\alpha_i^+, \alpha_i^-\}$, а також черговий растр $\{\alpha_r^+, \alpha_r^-\} = \{\alpha_r^+, \alpha_r^-\} \cap \{\alpha_i^+, \alpha_i^-\}$, тобто:

- якщо $(\alpha_r^+) > (\alpha_i^+)$, то $(\alpha_r^+) := (\alpha_i^+)$, у протилежному випадку (α_r^+) не змінюється;
- якщо $(\alpha_r^-) < (\alpha_i^-)$, то $(\alpha_r^-) := (\alpha_i^-)$, у протилежному випадку (α_r^-) не змінюється.

Таким чином здобувається новий черговий растр з множиною променів, для будь-якого з яких виповнюється критерій апроксимації і вони є можливими напрямками апроксимуючого відрізка.

Перевіряється умова сходження чергового растра. Якщо $(\alpha_r^+) \geq (\alpha_r^-)$, то растр ще існує і крок 3 повторюється. Якщо ж растр зійшовся $(\alpha_r^+) < (\alpha_r^-)$, то більше не існує променів, які б задовольняли критерію апроксимації, тоді здійснюється перехід до кроку 4.

Умовою виходу з кроку 3 до кроку 4 є також закінчення точок контуру, тобто останньою проаналізована початкова точка контуру.

4. Перевіряється умова, що остання можлива кінцева точка апроксимуючого відрізка є початковою точкою контуру. Якщо так, процедура апроксимації завершується. В протилежному випадку кінцева точка стає початковою точкою но-

вого чергового апроксимуючого відрізка ($z:=k, i:=k$) і здійснюється перехід до кроку 2.

Деякі аспекти практичної реалізації алгоритму. Для практичної реалізації алгоритму на сигнальних процесорах, щоб уникнути операцій типу *arcsin*, *arccos*, *arctg*, *arcctg* використовуються не кути α , а значення їх тригонометричних функцій ($\sin \alpha$, $\cos \alpha$), які в повній мірі задають напрямки променів на координатній площині. Далі оперуємо безпосередньо значеннями цих функцій не обчислюючи кутів. Обчислення ($\sin \alpha$, $\cos \alpha$) для прямої, що з'єднає початкову точку a_z з координатами (x_z, y_z) і точку a_i з координатами (x_i, y_i) , проводимо наступним чином:

$$\Delta x_i = x_i - x_z, \quad \Delta y_i = y_i - y_z; \quad \sin \alpha_i = \frac{\Delta y_i}{\sqrt{\Delta x_i^2 + \Delta y_i^2}}, \quad \cos \alpha_i = \frac{\Delta x_i}{\sqrt{\Delta x_i^2 + \Delta y_i^2}}.$$

Для обчислення растра, заданого як $\{(\sin \alpha_i^+, \cos \alpha_i^+), (\sin \alpha_i^-, \cos \alpha_i^-)\}$, використаємо відомі прості співвідношення:

$$\sin \Delta \alpha_i = \frac{\varepsilon}{\sqrt{\Delta x_i^2 + \Delta y_i^2}}, \quad \cos \Delta \alpha_i = \frac{\sqrt{\Delta x_i^2 + \Delta y_i^2 - \varepsilon^2}}{\sqrt{\Delta x_i^2 + \Delta y_i^2}};$$

$$\sin(\alpha_i^+) = \sin(\alpha_i + \Delta \alpha_i), \quad \cos(\alpha_i^+) = \cos(\alpha_i + \Delta \alpha_i);$$

$$\sin(\alpha_i^+) = \sin \alpha_i \cdot \cos \Delta \alpha_i + \cos \alpha_i \cdot \sin \Delta \alpha_i,$$

$$\cos(\alpha_i^+) = \cos \alpha_i \cdot \cos \Delta \alpha_i - \sin \alpha_i \cdot \sin \Delta \alpha_i;$$

$$\sin(\alpha_i^+) = \frac{\Delta y_i \cdot \sqrt{\Delta x_i^2 + \Delta y_i^2 - \varepsilon^2} + \Delta x_i \cdot \varepsilon}{\Delta x_i^2 + \Delta y_i^2},$$

$$\cos(\alpha_i^+) = \frac{\Delta x_i \cdot \sqrt{\Delta x_i^2 + \Delta y_i^2 - \varepsilon^2} - \Delta y_i \cdot \varepsilon}{\Delta x_i^2 + \Delta y_i^2}.$$

Відповідно:

$$\sin(\alpha_i^-) = \frac{\Delta y_i \cdot \sqrt{\Delta x_i^2 + \Delta y_i^2 - \varepsilon^2} - \Delta x_i \cdot \varepsilon}{\Delta x_i^2 + \Delta y_i^2},$$

$$\cos(\alpha_i^-) = \frac{\Delta x_i \cdot \sqrt{\Delta x_i^2 + \Delta y_i^2 - \varepsilon^2} + \Delta y_i \cdot \varepsilon}{\Delta x_i^2 + \Delta y_i^2}.$$

Для порівняння кутів скористаємося висновком, що $|(\alpha_i^+) - (\alpha_i^-)| < \pi$. Це ствердження справедливе і для $|(\alpha_r^+) - (\alpha_i)|$, $|(\alpha_r^-) - (\alpha_i)|$, $|(\alpha_r^+) - (\alpha_i^+)|$, $|(\alpha_r^-) - (\alpha_i^-)|$, $|(\alpha_r^+) - (\alpha_r^-)|$. Тобто порівняння кутів можна проводити по знаку функції *sin* різниці цих кутів.

Наприклад, для порівняння (α_r^+) і (α_i) обчислюється

$$\sin((\alpha_r^+) - (\alpha_i)) = \sin(\alpha_r^+) \cdot \cos(\alpha_i) - \cos(\alpha_r^+) \cdot \sin(\alpha_i).$$

Якщо обчислене значення більше або дорівнює 0, то відповідно $(\alpha_r^+) \geq (\alpha_i)$.

Перевірку $l(A_z, A_i) \in \varepsilon$ можна робити за обчисленою величиною $(\Delta x_i^2 + \Delta y_i^2 - \varepsilon^2)$, перевіряючи її за ознакою більше нуля.

В сигнальних процесорах операції множення з додаванням або відніманням є швидкими одноктактними операціями, тому перетворення з їх використанням займають мало часу, що значно впливає на швидкість алгоритму.

Виділення кутів у багатокутниках. У випадку коли апріорі відомо, що апроксимується контур багатокутника, то ставиться додаткова задача більш точного обчислення координат кутових точок. Така задача виникає при слідкуванні за плоским багатокутним об'єктом або маркером з відомими розмірами, з метою обчислення відстані до нього й орієнтації.

Вищенаведений алгоритм досить добре діє при відстеженні гострих кутів (за рахунок другої умови), але похибка в обчисленні координат кутових точок кутів, що дорівнюють або більші ніж $\pi/2$ може бути суттєвою і зростає при більших ε .

На рис. 2 пунктиром показано контур заданого багатокутника і лінією апроксимуючі відрізки, що сформовані при обході контуру за часовою стрілкою, починаючи з точки A .

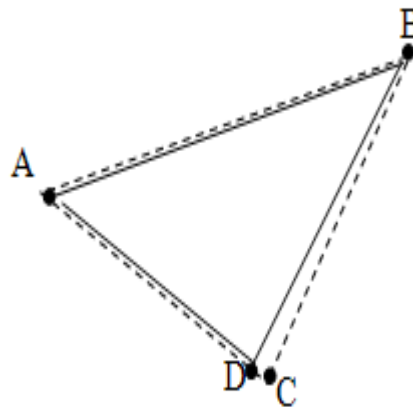


РИС. 2

В проміжних AB і DA лінії контуру і апроксимуючі відрізки збігаються, а апроксимуючий відрізок BD відхилився від лінії контуру BC , тому що згідно першій умові алгоритму всі точки контуру від B до D , у тому числі і точка C , знаходяться на відстані від апроксимуючого прямого відрізка BD , не більш ніж ε .

Таким чином за кінцеву точку апроксимуючого відрізка прийнято D , а не C , тобто кутами в апроксимуючому багатокутнику будуть A, B і D , а не A, B і C , що не відповідає дійсності.

Щоб цього не відбувалось, пропонується додаткова процедура для уточнення координат кутових точок. Після виділення кожного чергового апроксимуючого відрізка перевіряється відстань точок контуру, які знаходяться між його початковою і кінцевою точками, до прямої, що з'єднує ці точки. Точка, з максимальною відстанню близькою до ϵ , береться як кінцева для чергового і початкова для наступного етапу апроксимації.

Наприклад, якщо виділено черговий апроксимуючий відрізок BD , кутова точка C , буде знаходитися на максимальній відстані від BD . Вона буде взята за кінцеву точку і з неї почнеться виділення наступного апроксимуючого відрізка, а черговим стане BC замість BD .

Обчислення відстаней досить просто можна проводити наступним чином. Перемістимо і повернемо сегмент контуру між точками B і D так, що точка B буде знаходитись у центрі координатної сітки, а D лежатиме на осі абсцис. Тоді координати точок на осі ординат, взяті по модулю, будуть відстанями до прямої BD .

Функцією обчислення відстані для i -тої точки контуру буде:

$$h_i = |(y_i - y_b) \cdot \cos \gamma + (x_i - x_b) \cdot \sin \gamma|,$$

де

$$\sin \gamma = \frac{y_D - y_B}{\sqrt{(y_D - y_B)^2 + (x_D - x_B)^2}}, \quad \cos \gamma = \frac{x_D - x_B}{\sqrt{(y_D - y_B)^2 + (x_D - x_B)^2}},$$

а $(x_B, y_B), (x_D, y_D), (x_i, y_i)$ – відповідно координати точок B, D , а також точок аналізованого сегменту контуру.

Реалізація алгоритму апроксимації в інтелектуальній відеокамері. Алгоритм апроксимації за наведеним вище методом був реалізований, як один з компонентів програмного забезпечення для аналізу й обробки зображень в інтелектуальній відеокамері (ІВК) створеній в Інституті кібернетики імені В.М. Глушкова НАН України. Далі на рис. 3, 4, 5 показані результати роботи демонстраційної програми по виділенню контурів об'єктів зображення, апроксимації їх багатокутниками і ідентифікації фігур типу квадрат та ромб за критеріями кількості кутів і співвідношення їх сторін і діагоналей. Вся обробка виконувалась в інтелектуальній відеокамері, а результати передавались у комп'ютер і відображались на екрані монітора.

На рис. 3 показано знімок деяких об'єктів з виділеними контурами та переведеними в векторну форму для подальшої обробки. Видно, що сторони фігур мають флуктуації і представляють собою лінії з деякими відхиленнями.

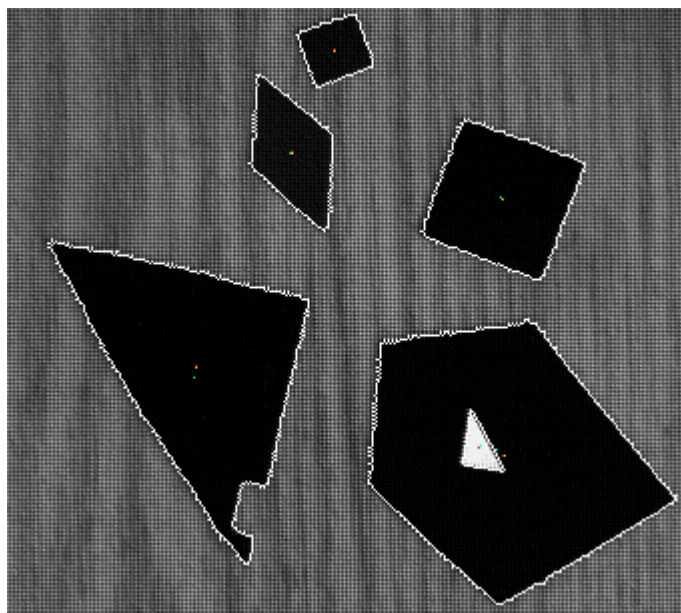


РИС. 3

Після апроксимації, з заданою невеликою похибкою, в ІВК обчислені координати кутів фігур. На рис. 4 показано відображення фігур та їх контурів, отриманих за рахунок з'єднання прямими лініями точок кутів з координатами, отриманими із ІВК.

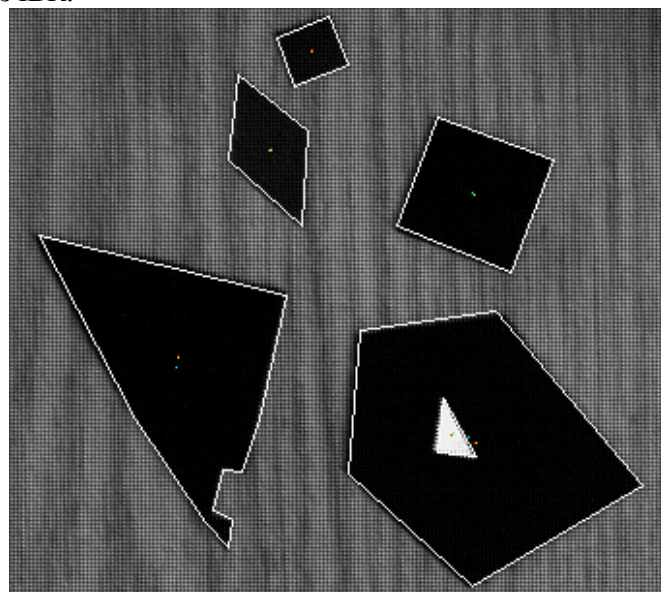


РИС. 4

На рис. 5 показано вікно з відображенням результатів обробки зображень і ідентифікації фігур, проведеними в ІВК і переданими в комп'ютер. У вікні розташовані значення різних характеристик аналізованих об'єктів, у тому числі, кількість кутів (кол-во пикс. периметра) апроксимованих контурів фігур. Координати точок контурів і кутів передавались у комп'ютер окремими масивами.

	kvadrat	romb	kvadrat	kvadrat
Индекс объекта	1,0000h	1,0000h	1,0000h	1,0000h
Длина периметра	136	241	314	641
S периметра	945	2037	4205	9858
S объекта	945	2037	4204	9858
Кв.расст. min/max	334	685	717	1636
Адрес БУЗУ периметра	9300h 0000h	9300h 01D8h	9300h 0504h	9300h 09...
Кол-во пикс. периметра	4	4	4	10
Расст. Ц. пер.-Ц. объекта	0	0	0	0
Коорд. окна [min,max]	35, 192, 71, 230	65, 168, 143, 210	87, 254, 167, 336	147, 68, ...
Центр массы перим.	i=52,25 j=211,125	i=103,5 j=189,3125	i=125,6875 j=294,4375	i=215,5625
Центр массы площ.	i=52,3125 j=211,25	i=103,25 j=190,125	i=126,625 j=294,8125	i=210,125
Центр массы объекта	i=52,5 j=211,4375	i=103,75 j=190,1875	i=126,875 j=295,125	i=210,6875
Коорд.т. max расст. от Ц.	i=71 j=202	i=143 j=206	i=109 j=336	i=309 j=16...
Коорд.т. min расст. от Ц.	i=39 j=209	i=117 j=177	i=97 j=303	i=228 j=11...
Корень кв. [Sперим., Sobj]	Sperim=30,75 Sobj=30,75	Sperim=45,125 Sobj=45,12	Sperim=64,875 Sobj=64,81	Sperim=99...

РИС. 5

Висновки. Реалізація запропонованого методу апроксимації замкнених контурів зображень багатокутниками в ІВК та проведені експерименти з різними типами фігур і різними значеннями похибок апроксимації підтвердила його працездатність та ефективність.

Запропонований метод і алгоритм можуть бути застосовані, зокрема, для візуального контролю геометричних розмірів продукції у системах технічного зору різних галузей промисловості.

1. Мельник Э.И., Выделение прямых и дуг окружностей в технических чертежах // Автоматизация обработки и распознавания изображений. – Минск: ИТК АНБ, 1995. – С. 147 – 154.
2. Бутаков В.А., Островский В.И., Фадеев И.Л. Обработка изображений на ЭВМ. – М.: Радио и связь, 1987. – 240 с.

Одержано 25.10.2013