

УДК 004.9

## ПОРОГОВА СЕГМЕНТАЦІЯ ДАНИХ ДИСТАНЦІЙНОГО ЗОНДУВАННЯ ЗЕМЛІ

Р. Сулимко, Р. Шувар

*Львівський національний університет імені Івана Франка  
вул. Драгоманова, 50, 79005 Львів, Україна  
rsulymko@ukr.net*

Розглянуто методику сегментації растрових зображень дистанційного зондування Землі високої роздільної здатності з використанням кольорових і текстурних характеристик зображень. Розроблено програмну реалізацію запропонованої методики. Проаналізовано вплив розмірів вікон та значення порогу для коефіцієнта відхилення на точність сегментації. Досліджено ефективність запропонованої методики на різних елементах зображень.

*Ключові слова:* сегментація, текстурна характеристика, дистанційне зондування Землі.

Доступність даних дистанційного зондування Землі високої роздільної здатності зумовила необхідність пошуку ефективних алгоритмів перетворення вхідних даних для геоінформаційних систем з растрового у векторний формат. Для вирішення цього завдання, зазвичай, проводять сегментацію, щоб отримати бінарні шари, які відповідають окремим елементам вхідного зображення (гідрографія, ліси, дороги тощо).

Для сегментації зображень сьогодні запропоновано багато алгоритмів, які можна розділити на такі підмножини [6]:

- 1) методи порогової сегментації;
- 2) методи, засновані на кластеризації;
- 3) методи з використанням гістограм;
- 4) виділення країв;
- 5) методи розростання областей;
- 6) методи розрізу графа;
- 7) сегментація методом водоподілу;
- 8) сегментація за допомогою моделі.

Алгоритми порогової обробки зображень досить прості в реалізації. Поріг – це ознака (властивість), яка допомагає розділити шуканий сигнал на класи. Операція порогового поділу полягає в порівнянні функції (наприклад, функція може повертати яскравість пікселя) для кожного пікселя зображення із заданим значенням порогу. Алгоритми з цієї групи можуть давати неточні результати, особливо за високого рівня шуму на зображенні чи у випадку погано обраного значення порогу [7]. Методи порогової сегментації потребують модифікації та додаткової обробки зображень.

Висока роздільна здатність даних дистанційного зондування Землі дає змогу визначити текстурні характеристики елементів на зображенні. Ми запропонували методику

порогової сегментації растрових зображень, яка, крім кольорових характеристик, враховує їхні текстурні характеристики.

В основі запропонованого підходу сегментації є використання швидкого дискретного двовимірного Фур'є-перетворення для аналізу зображень. Як вхідні дані для Фур'є-перетворення використано величини градацій інтенсивностей складових каналів кольору пікселів опорного та поточного зображень.

Обчислюють середньоквадратичне відхилення амплітуд гармонік поточного зображення щодо амплітуд гармонік опорного зображення з урахуванням внеску кожної складової кольору та вагомості кожної гармоніки спектра опорного зображення. На підставі аналізу коефіцієнта відхилення від порогового значення приймають рішення про сегментацію. Для опрацювання всього вхідного зображення в циклі виконують його поступовий аналіз за допомогою поточного вікна.

Як вхідні зображення використовували космічні знімки Землі високої роздільної здатності, зняті супутником Digital Globe Cnes/spot Image.

Проект розроблений в операційній системі Linux (OpenSUSE 12.2). Графічною оболонкою для програмування обрано Code::Blocks — вільне кросплатформне середовище з відкритим кодом [3].

Для роботи з растровими зображеннями використано бібліотеку FreeImage, що опрацьовує растрові зображення форматів PNG, BMP, JPEG, TIFF. Бібліотека FreeImage забезпечує легкий доступ до bitmap-компонентів та основні маніпуляції з растровими зображеннями [4]. Для роботи бібліотеки FreeImage використано необхідні внутрішні функції управління бібліотекою (FreeImage\_Initialise, FreeImage\_DeInitialise) в коді програми:

```
#ifdef FREEIMAGE_LIB
    FreeImage_Initialise();// Ініціалізація бібліотеки
#endif // FREEIMAGE_LIB
#ifdef FREEIMAGE_LIB
    FreeImage_DeInitialise();// Очистка виділених ресурсів у бібліотеці FreeImage.
#endif // FREEIMAGE_LIB
```

Розглянемо, як приклад, задачу сегментації лісових насаджень на вхідному зображенні (рис. 1).

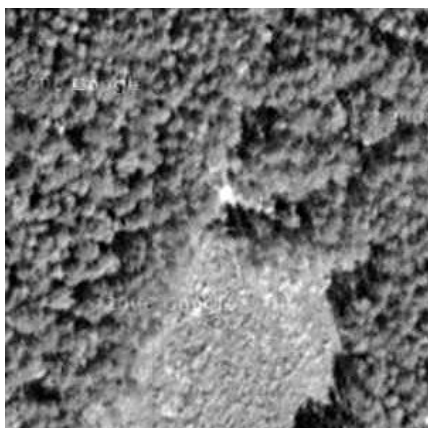


Рис. 1. Вхідне зображення

За допомогою функції `FreeImage_Load` реалізовано завантаження вхідного зображення:

```
FIBITMAP *inz = FreeImage_Load(FIF_BMP, "7.bmp", BMP_DEFAULT); //зчитуємо вхідне зображення
```

Ця функція декодує растрове зображення, виділяє для нього пам'ять, а потім повертає його як `FIBITMAP`. Перший параметр визначає тип растрового формату, що має бути завантажений. Другий параметр повідомляє `FreeImage`, який файл він повинен декодувати. Останній параметр використовують для зміни поведінки або ввімкнення функції в растровому плагіні. Кожен модуль має свій власний набір параметрів [4].

На наступному етапі виконано зчитування `bitmap`-компонентів зображення, що потребувало доступу до пікселів. Функції `FreeImage` для доступу до пікселів надають простий спосіб зчитувати, записувати й опрацьовувати піксель за пікселем з `FIBITMAP`-даних.

Для того щоб отримати колір пікселя з зображення у позиції  $(x, y)$ , бібліотека `FreeImage` використовує функцію `GetPixelColor`. Параметр "x" – місце розташування пікселя в горизонтальному напрямі, а параметр "y" є позицією пікселя у вертикальному напрямі:

```
FreeImage_GetPixelColor(inz,x,y,&pixcol);
int r = pixcol.rgbRed; int g = pixcol.rgbGreen; int b = pixcol.rgbBlue;
```

З використанням функції `GetPixelColor` сформовано три двовимірні масиви  $r, g, b$ , елементами яких є величини градацій інтенсивностей складових каналів кольору кожного пікселя (`pixcol.rgbRed`; `pixcol.rgbGreen`; `pixcol.rgbBlue`).

Для отримання кількісних характеристик елементів зображення обираємо на вхідному зображенні сегмент, який найліпше характеризує ділянки, які потрібно виокремити. Колірні та текстурні характеристики цього сегмента приймають за опорні (еталонні) в разі прийняття рішення про виділення подібних сегментів вхідного зображення. Відповідно до розмірів вікна опорного сегмента ( $n_x * n_y$ ) формуються три підматриці з величинами градацій інтенсивностей складових каналів кольору (Red, Green, Blue)  $E_{r,g,b}$ .



Рис. 2. Опорне зображення

Для реалізації алгоритму швидкого дискретного двовимірного перетворення Фур'є використано бібліотеку `FFTW`. Бібліотека `FFTW` – це набір швидких C-підпрограм для обчислення дискретного перетворення Фур'є (ДПФ) складних даних, реальних даних, парно-непарних або симетричних реальних даних [5].

Функцією (`fftw_execute(plan_forward)`) виконано пряме дискретне двовимірне перетворення Фур'є, на вхід якого подано дані з підматриць  $E_{r,g,b}$  опорного зображення:

```
plan_forward = fftw_plan_dft_r2c_2d ( nx, ny, in, out, FFTW_ESTIMATE );
fftw_execute ( plan_forward );
```

Отримано для опорного зображення три матриці зі значеннями гармонік для кожного каналу кольору  $F(E_{r,g,b})$ . Для кожної гармоніки опорного зображення обчислено її ваговий коефіцієнт значимості.

$$\alpha_{ij} = \frac{|F_{ij}^{(1)}|}{\left(\sum_{i,j=1}^n |F_{ij}^{(1)}|\right) / n} \quad (1)$$

де  $\alpha_{ij}$  – ваговий коефіцієнт значимості гармоніки відповідного каналу кольору;  $F_{ij}^{(1)}$  – значення гармоніки опорного зображення відповідного каналу кольору;  $n$  – кількість гармонік.

Для визначення внеску кожної складової кольору (червоного, зеленого, синього) обчислюємо розподіл кольорів для опорного зображення, вводимо коефіцієнт значимості величин складових каналів кольору:

$$\begin{aligned} \sum r &= \frac{\sum_{i,j=1}^n r_{ij}}{n}; \quad \sum g = \frac{\sum_{i,j=1}^n g_{ij}}{n}; \quad \sum b = \frac{\sum_{i,j=1}^n b_{ij}}{n}; \\ \kappa_r &= \frac{\sum r}{(\sum r + \sum g + \sum b)}; \\ \kappa_g &= \frac{\sum g}{(\sum r + \sum g + \sum b)}; \\ \kappa_b &= \frac{\sum b}{(\sum r + \sum g + \sum b)} \end{aligned} \quad (2)$$

де  $r_{ij}$  – значення червоної складової кольору кожного пікселя;  $g_{ij}$  – значення зеленої складової кольору кожного пікселя;  $b_{ij}$  – значення синьої складової кольору кожного пікселя;  $\kappa_r$  – коефіцієнт значимості червоної складової кольору;  $\kappa_g$  – коефіцієнт значимості зеленої складової кольору;  $\kappa_b$  – коефіцієнт значимості синьої складової кольору;  $n$  – кількість пікселів.

Аналіз вхідного зображення передбачає формування поточного вікна, розміри якого відповідають розмірам вікна опорного зображення. Вікно поточного зображення проходить попіксельно усе вхідне зображення, рухаючись з верхнього лівого кута в нижній правий. За допомогою поточного вікна із вхідного зображення сформовані поточні підматриці  $W_{r,g,b}$ , які використовують для порівняння з опорним вікном. Для підматриць  $W_{r,g,b}$  поточного зображення виконують перетворення Фур'є. Отримано три матриці поточного зображення зі значеннями гармонік для кожного кольору  $F(W_{r,g,b})$ .

Для кожної гармоніки поточного зображення обчислюють середньоквадратичне відхилення щодо відповідної гармоніки опорного зображення з відповідної матриці з урахуванням коефіцієнта вагомості гармонік та коефіцієнта значимості величин складових каналів кольору:

$$\beta = \kappa \frac{\sum_{i,j=1}^n \alpha_{ij} (F_{ij}^{(1)} - F_{ij}^{(2)})^2}{n} \quad (3)$$

де  $\beta$  – середньоквадратичне відхилення амплітуд гармонік поточного та еталонного зображень відповідного каналу кольору;  $\alpha_{ij}$  – ваговий коефіцієнт значимості гармоніки поточного зображення відповідного каналу кольору;  $F_{ij}^{(1)}$  – значення гармоніки поточного зображення відповідного каналу кольору;  $F_{ij}^{(2)}$  – значення гармоніки еталонного зображення відповідного каналу кольору;  $n$  – кількість пікселів;  $\kappa$  – коефіцієнт значимості відповідної складової кольору.

Відповідно до порогового значення коефіцієнта відхилення, виділяємо подібні сегменти. Приймаємо рішення: сегменти, середньоквадратичне відхилення яких менше від порогового значення, приєднати до області зображення, яку потрібно виділити. З відповідних сегментів формуємо вихідне чорно-біле зображення.

Після порівняння спектрів опорного і поточного зображень вікно поточного зображення зміщується вправо на 1 піксель, після чого програма повертається до етапу формування поточних підматриць  $W_{r,g,b}$  та виконує знову всі етапи обробки та порівняння.

Якщо необхідно створити нове растрове зображення в пам'яті з нуля, без завантаження готового растру з файлу, застосовують функцію `FreeImage_Allocate`. Ця функція використовує як параметри ширину, висоту та BPP-параметр, який визначає глибину кольору зображення, а повертає `FIBITMAP`. Три додаткові параметри (`red_mask`, `green_mask` і `blue_mask`) використовують, щоб задати функції `FreeImage` біт розташування кольорних компонент у растрі. Функція `FreeImage_Allocate` виділяє порожнє растрове зображення, наприклад, растрове зображення, яке повністю заповнене нулями. Нулі в растровому зображенні зазвичай інтерпретуються як чорний колір [4].

Для відображення результатів з використанням функції `FreeImage_Allocate` сформовано порожнє зображення.

```
out1 = FreeImage_Allocate(w,h,24);
```

Для формування вихідного зображення за допомогою функції `FreeImage_SetPixelColor` подібні сегменти забарвлюються в чорний колір (`pixcol.rgbBlue=0; pixcol.rgbGreen=0; pixcol.rgbRed=0`), а всі інші – у білий (`pixcol.rgbBlue=255; pixcol.rgbGreen=255; pixcol.rgbRed=255`).

```
if (dil < 60 ) {pixcol.rgbBlue=0; pixcol.rgbGreen=0; pixcol.rgbRed=0; pixcol.rgbReserved=1;
FreeImage_SetPixelColor(out1,x,y, &pixcol);}
else {pixcol.rgbBlue=255; pixcol.rgbGreen=255; pixcol.rgbRed=255; pixcol.rgbReserved=1;
FreeImage_SetPixelColor(out1,x,y, &pixcol);}
```

Функцією `FreeImage_Save` збережено чорно-біле растрове сегментоване зображення:

```
const char *output_filename1 = "out.jpg";
FREE_IMAGE_FORMAT out_fif = FreeImage_GetFIFFromFilename(output_filename1);
FreeImage_Save(out_fif, out1, output_filename1, 0);
```

Ця функція зберігає раніше завантажений `FIBITMAP` у файл. Перший параметр визначає тип растрового зображення. Наприклад, коли передається `FIF_BMP`, зберігається файл `BMP`. Інший параметр – це ім'я растрового зображення, що зберігається. Якщо файл уже існує, то він буде перезаписаний. Останній параметр використовується для зміни поведінки чи ввімкнення функції в растровому файлі.

Якщо виконувати сегментацію за кожною складовою каналів кольору вхідного растрового зображення (червоний, синій, зелений), то результати будуть незадовільними.

Тому для зменшення похибки сегментацію провели з урахуванням коефіцієнта вагомості гармонік та коефіцієнта значимості величин складових каналів кольору.

Розмір вікна опорного зображення залежить від текстури: чим дрібніша текстура, тим менший розмір вікна. У випадку вибору замалого розміру вікна опорного зображення втрачається структурна характеристика (текстура) зображення, а для завеликого розміру вікна збільшується похибка обчислень. Значення порога коефіцієнта відхилення залежить від розміру опорного вікна: менше вікно – менший поріг, зі збільшенням вікна збільшується поріг.

На рис. 3–6 зображено приклади сегментації елементів зображення з різними текстурами. Затемнені ділянки відображають сегментовані області, широкую лінію проводять вручну за критерієм того, як людське око сприймає межі поділу областей.

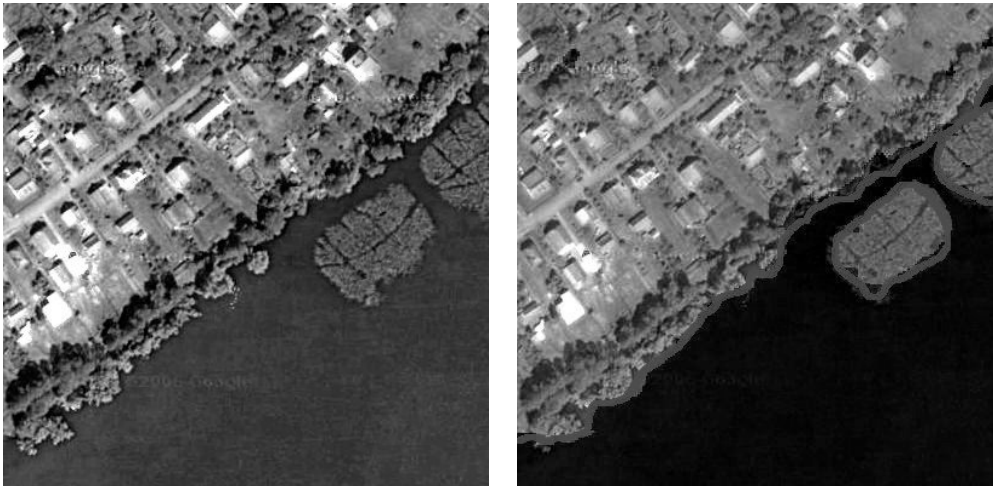


Рис. 3. Виділення озер методом порогової сегментації.

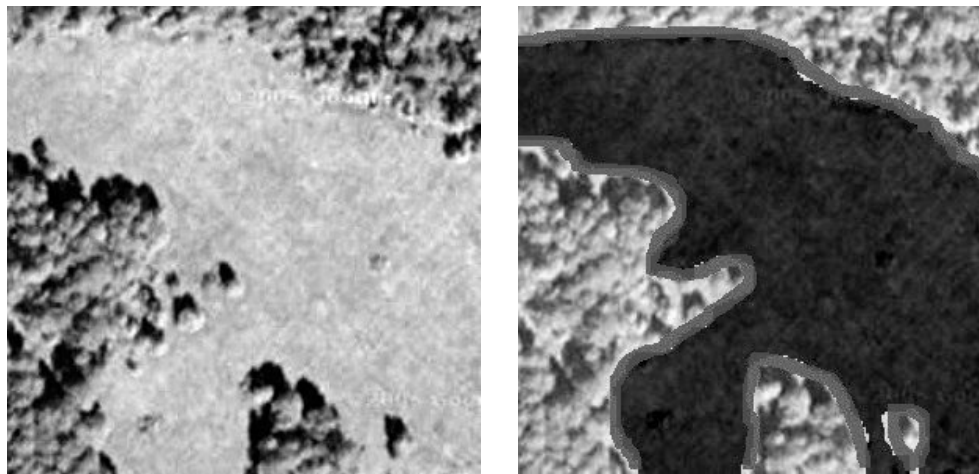


Рис. 4. Виділення галявин методом порогової сегментації

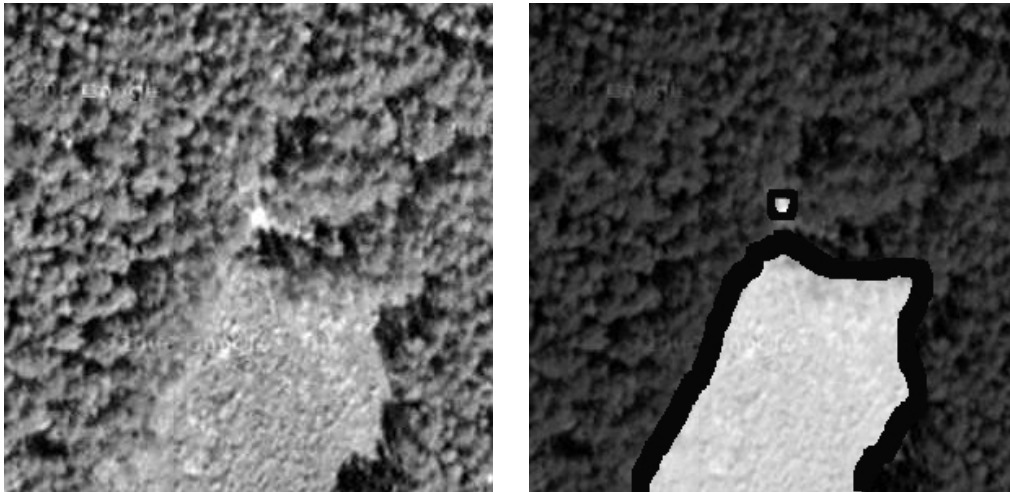


Рис. 5. Виділення лісу методом порогової сегментації

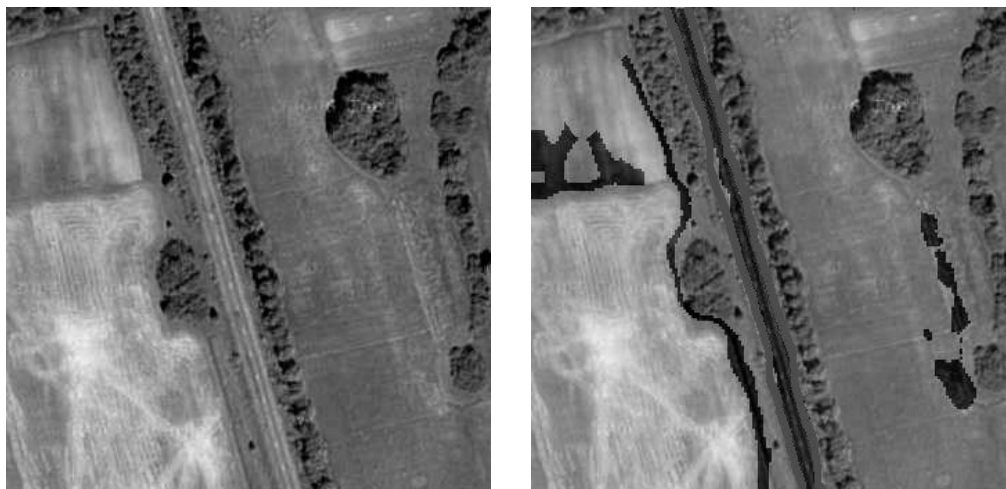


Рис. 6. Виділення доріг методом порогової сегментації.

Для задачі сегментації озер з'ясовано, що найефективніше програма виконується у вікні опорного зображення розміром  $20 \times 20$  px з порогом 10 (див. рис. 3), що зумовлено дрібною текстурою.

Задача сегментації галявин найефективніше працює з вікном опорного зображення розміром  $20 \times 20$  px з порогом 10 з огляду на дрібну текстуру (див. рис. 4).

У задачі сегментації лісу використовували вікно опорного зображення розміром  $50 \times 50$  px з порогом 60 (див. рис. 5). Розмір вікна для виділення лісу збільшено, оскільки за меншого вікна втрачається структурна характеристика лісу.

Для задачі сегментації доріг використовували вікно опорного зображення розміром  $20 \times 20$  px з порогом 10 (див. рис. 6). Розмір вікна опорного зображення зумовлений розмірами дороги.

Отже, запропоновано підхід до аналізу растрових даних, в основі якого – використання прямого швидкого дискретного двовимірного Фур'є-перетворення. На підставі порівняння спектрів опорного та поточного елементів зображення виконано сегментацію растрових зображень.

Досліджено також вплив розміру вікна опорного зображення на ефективність сегментації та ефективність підбору порогового значення коефіцієнта відхилення різних текстур зображення.

- 
1. *Барталев С. А.* Анализ возможностей применения методов сегментации спутниковых изображений для выявления изменений в лесах / С. А. Барталев, Т. С. Ховратович // Современные проблемы дистанционного зондирования Земли из космоса – 2011. – Т. 8, – № 1. – С. 44–62.
  2. *Сулимко Р. Т.* Реалізація проекту порогової сегментації зображень в IDE CodeBlocks з використанням бібліотек FFTW та FreeImage / Р. Т. Сулимко, Р. Я. Шувар // Третя міжнар. наук.-практ. конф. “FOSS Lviv 2013” : збірник наукових праць, 18–21 квітня 2013 р. – Львів : приватний підприємець Сорока Т. Б., 2013. – С. 169–171.
  3. Code::Blocks – [Електронний ресурс] – Режим доступу: <http://www.codeblocks.org/>
  4. The FreeImage Project – [Електронний ресурс] – Режим доступу: <http://freeimage.sourceforge.net/>
  5. FFTW Home Page – [Електронний ресурс] – Режим доступу: <http://freeimage.sourceforge.net/>
  6. Сегментація зображень-Вікіпедія – [Електронний ресурс] – Режим доступу: <http://uk.wikipedia.org/wiki/>
  7. *Поршнев С. В., Левашкина А. О.* Универсальная классификация алгоритмов сегментации изображений. – [Електронний ресурс] – Режим доступу: <http://jurnal.org/articles/2008/inf23.html>
  8. *Комаричев М. Е., Зори С. А.* Сегментация изображений на параллельных вычислительных системах – [Електронний ресурс] – Режим доступу: <http://masters.donntu.edu.ua/2012/fknt/komarychev/library/article1.htm>



---

**DATA SEGMENTATION THRESHOLD  
OF THE EARTH REMOTE SONDAGE IMAGES****R. Sulymko, R. Shuvar**

*Ivan Franko National University of Lviv,  
50 Drahomanov St., UA-79005 Lviv, Ukraine  
rsulymko@uk.net*

In this paper, the technique of high resolution bitmap segmentation is proposed. The result of it is a draft raster image segmentation, which consider their textural characteristics in addition to color characteristics into. The operation of the program and examples of processed bitmaps are described. The effectiveness of various threshold value selection for the coefficient of deviation of different textures in images is shown. The effectiveness of the implemented project is analysed on different image textures.

*Key words:* segmentation, texture characteristics, remote sondage of the Earth.

**ПОРОГОВАЯ СЕГМЕНТАЦИЯ ДАННЫХ  
ДИСТАНЦИОННОГО ЗОНДИРОВАНИЯ ЗЕМЛИ****Р. Сулимко, Р. Шувар**

*Львовский национальный университет имени Ивана Франко  
ул. Драгоманова, 50, 79005 Львов, Украина  
rsulymko@ukr.net*

Рассмотрено методику сегментации растровых изображений высокого разрешения. Разработано проект сегментации растровых изображений, который, помимо цветовых характеристик, учитывает их текстурные особенности. Описано принципы работы и приведено примеры разработки программы обработки растровых изображений. Проанализировано эффективность подбора порогового значения для коэффициента отклонения различных текстур изображения. Проанализировано эффективность работы реализованного проекта на разных текстурах изображения.

*Ключевые слова:* сегментация, текстурная характеристика, дистанционное зондирование Земли.

Стаття надійшла до редколегії 10.04.2013

Прийнята до друку 14.05.2013