Сулимко Р. Т., Шувар Р. Я.
Львівський національний університет імені Івана Франка

Реалізація проекту порогової сегментації зображень в IDE CodeBlocks з використанням бібліотек fftw та freeimage

Сьогодні набули широкого розвитку галузі науки і техніки, у яких використовують системи обробки інформації з використанням даних у цифровому вигляді.

Одним з основних типів інформаційних систем, що використовують цифрові дані, є геоінформаційні системи (ГІС).

Використання зображень в ГІС зумовлює опрацювання великих обсягів растрових даних, що в свою чергу зумовлює необхідність розробки нових та використання для цього більш ефективних обчислювальних систем аналізу візуальної інформації.

У цій роботі розглянуто використання вільного програмного забезпечення для аналізу растрових зображень.

Авторами попередньо було реалізовано проект порогової сегментації растрових зображень ДЗЗ високої роздільної здатності. Проект був реалізований в середовищі Microsoft Visual Studio на мові C++. Проте у зв’язку з необхідністю використання легального програмного забезпечення виникло завдання реалізації цієї задачі сегментації на вільному програмному забезпеченні.

- Платформою для розробки проекту було обрано операційну систему OpenSUSE 12.2.
- Середовищем для програмування було обрано Code::Blocks.

Code::Blocks — вільне кросплатформенне середовище з відкритим кодом. Ця система використовується як для написання невеликих проектів для вбудованих додатків, так і для програмування додатків для PC під Windows, Linux і MacOs. Маючи компактне ядро, вона може масштабувати і розширити свій функціонал за допомогою безлічі додаткових модулів. Підтримує мови програмування C, C++ та інші. Має компактну та інтуїтивно зрозумілу структуру меню, що забезпечує швидке налаштування середовища.

На першому етапі розробки програми було обрано та під'єднано необхідні бібліотеки:

- FreeImage - для опрацювання растрових зображень;
- FFTW - для реалізації алгоритму швидкого дискретного двовимірного перетворення Фур'є.

FreeImage є бібліотекою з відкритим кодом, розрахованою для розробників, яким необхідно опрацьовувати растрові зображення, таких форматів як PNG, BMP, JPEG, TIFF. FreeImage може використовуватися на багатьох мовах, включно з C, C++, VB, C#, Delphi, Java.

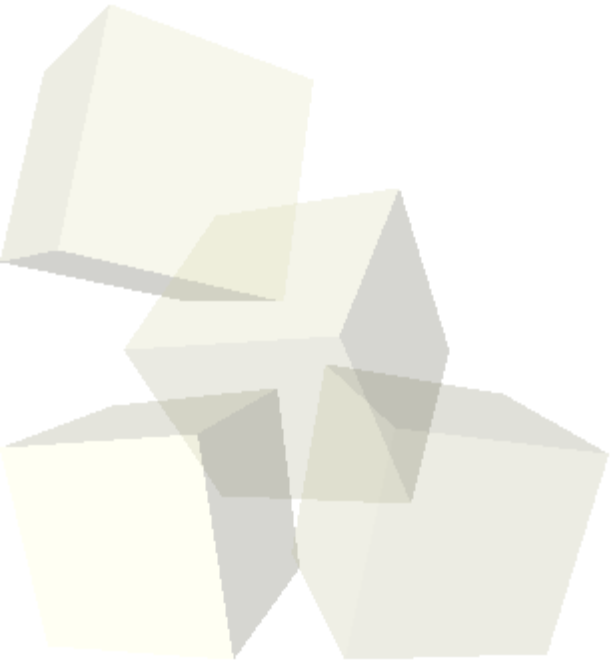
FreeImage дає змогу: завантаження і збереження багато типів растрових зображень, легкий доступ до bitmap-компонентів, конвертування растрового зображення з одного типу до іншого, основних маніпуляцій з растровими зображеннями, регулювання яскравості та контрастності, альфа-компонування та альфа-змішування.

FFTW-бібліотека являє собою набір швидких С підпрограм для обчислення дискретного перетворення Фур'є (ДПФ).

• FFTW обчислює ДПФ складних даних, реальних даних, парно-непарних або симетричних реальних даних.

• введення даних може мати довільну довжину.

• FFTW підтримує довільні багатовимірні дані.



FFTW

Для роботи бібліотеки FreeImage було викликано необхідні внутрішні функції управління бібліотекою (FreeImage_Initialise, FreeImage_DeInitialise) в коді програми.

За допомогою функції FreeImage_Load реалізовано завантаження вхідного зображення. В якості вхідних зображень використовувались космічні знімки Землі високої роздільної здатності.

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

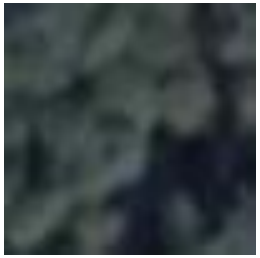
Вхідне зображення

На наступному етапі, для подальшого аналізу та обробки зображень, здійснено зчитування bitmap-компонентів зображення, що потребувало доступу до пікселів. Для того щоб отримати колір пікселя з 16 -, 24 - або 32-бітових зображень у позиції (x, y) бібліотека FreeImage використовує функцію GetPixelColor.

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

Використовуючи функцію 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, колірні та текстурні характеристики якого будуть опорними при прийнятті рішення про виділення подібних сегментів вхідного зображення. Відповідно до розмірів опорного сегмента (nx*ny) формуються три підматриці з величинами градацій інтенсивностей складових каналів кольору (Red, Green, Blue).



Еталонне зображення

Було сформовано поточне вікно, розміри якого відповідали розмірам опорного вікна.
За допомогою поточного вікна з вхідного зображення формувалися поточні масиви r, g, b, які використовувалися для порівняння з опорним вікном.

Функцією (fftw_execute (plan_forward)) здійснено пряме дискретне двовимірне перетворення Фур'є , на вхід якого було подано дані з масивів r, g, b опорного та поточного зображень.

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

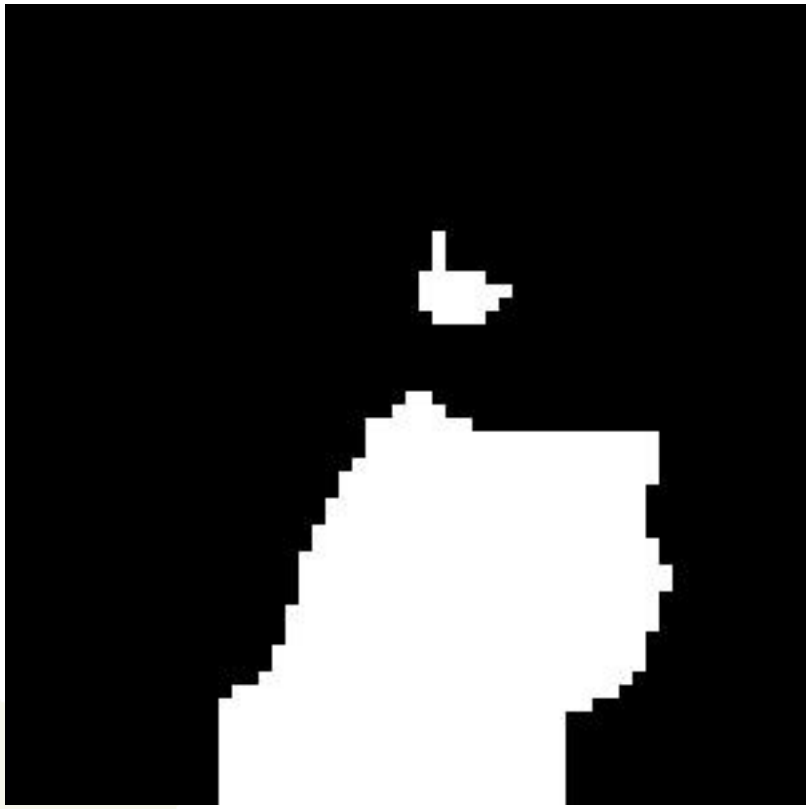
Отримано по три матриці (out) опорного та поточного зображень із значеннями гармонік для кожного кольору. Для кожної гармоніки обчислюємо її ваговий коефіцієнт значимості. Для визначення внеску кожної складової кольору (червоного, зеленого, синього), обраховуємо розподіл кольорів для опорного зображення, вводимо коефіцієнт значимості величин складових каналів кольору.

Для кожної матриці опорного і поточного зображень обчислюємо середньоквадратичне відхилення амплітуд гармонік, з врахуванням коефіцієнта вагомості гармонік та коефіцієнту значимості величин складових каналів кольору.Для відображення результатів, використовуючи функцію FreeImage_Allocate сформовано порожнє зображення.

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

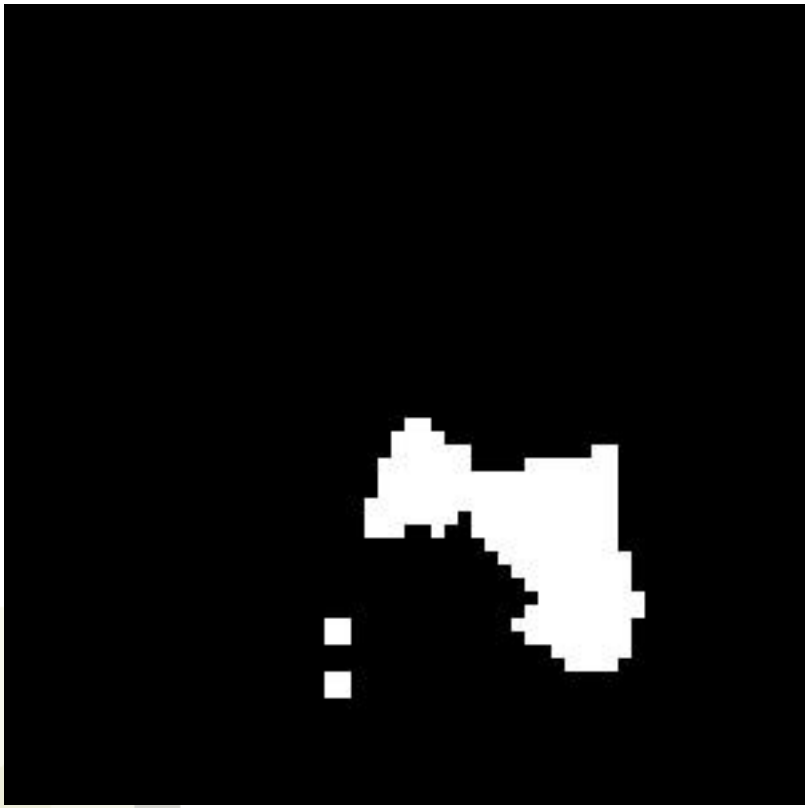
Відповідно до порогового значення коефіцієнта відхилення виділяємо подібні сегменти. З відповідних сегментів формуємо вихідне чорно-біле зображення.

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

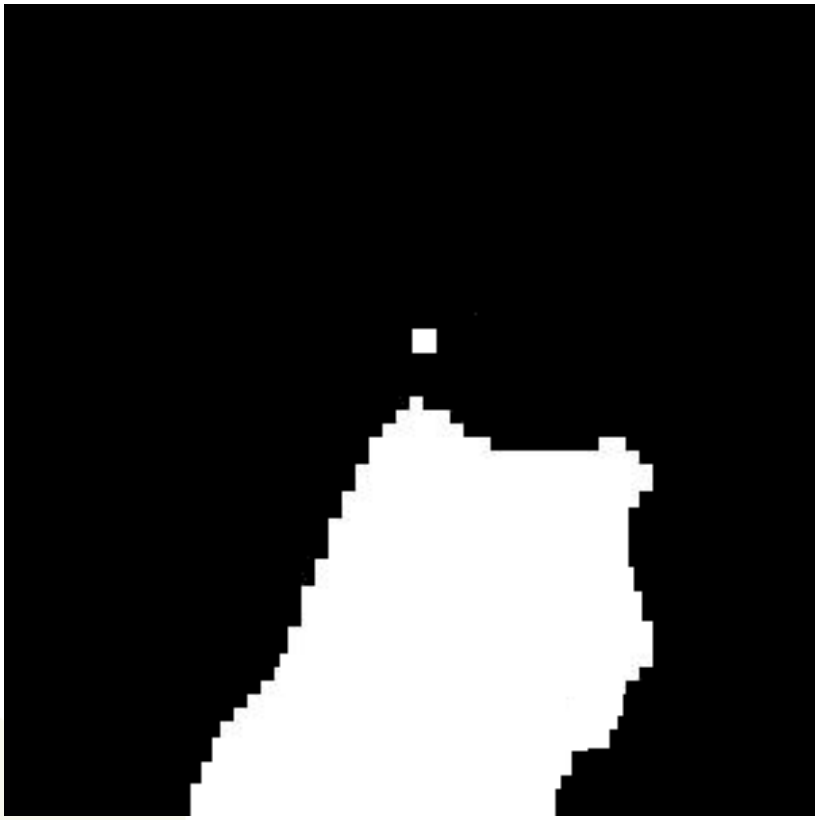


Порівняння по червоному компоненту кольору

Порівняння по зеленому компоненту кольору



Порівняння по синьому компоненту кольору



Порівняння з врахуванням коефіцієнта вагомості гармонік та коефіцієнту значимості величин складових каналів кольору.

Реалізація цього проекту підтверджує можливість розробки складних програмних продуктів на вільному програмному забезпеченні.

Дякую за увагу!