

Janusz Kosakowski

OBRAZY ZDJĘĆ LOTNICZYCH W POSTACI CYFROWEJ JAKO WARSTWA INFORMACYJNA W GIS WSPIERAJĄCYM GOSPODARKĘ PRZESTRZENIĄ W GMINIE. OPIS KONCEPCJI PROGRAMU W ŚRODOWISKU WINDOWS.

Referat opisuje koncepcję programu komputerowego integrującego cyfrowe obrazy fotogrametrycznych zdjęć lotniczych z przestrzennym aspektem własności gruntów. Aplikacja taka powinna zawierać narzędzia do obsługi obrazów rastrowych, grafiki wektorowej oraz bazy danych. Z uwagi na geometrię zdjęć (rzut środkowy), niezbędne też jest rozwiązanie zagadnień związanych z przekształceniami rzutowymi oraz orientacją wzajemną i bezwzględną stereogramów.

1. Wprowadzenie

Gospodarka przestrzenna wymaga podejmowania działań w takich dziedzinach jak kataster nieruchomości, planowanie przestrzenne, ochrona środowiska itd... . Dzisiaj trudno sobie wyobrazić wykonywanie związanych z tym zadań bez komputerów i przetwarzania dużej ilości informacji. Odpowiednim do tego środowiskiem jest środowisko GIS.

Według Yue-Hong Chou: „GIS jest zorganizowanym zestawem sprzętu, oprogramowania, danych geograficznych oraz personelu, zaprojektowanym do efektywnego zbierania, przechowywania, uaktualniania, przetwarzania, analizowania i prezentowania wszelkich form informacji odniesionej geograficznie” [Chou Y., 1997].

Podstawowym zadaniem GIS jest tworzenie informacji na podstawie danych. W związku z tym system taki składa się zwykle z trzech modułów: grafiki rastrowej, grafiki wektorowej i bazy danych, wraz z narzędziami właściwymi dla tych form danych.

Obecnie na rynku istnieje dosyć szeroka oferta środowisk GIS, które przy różnym nakładzie środków umożliwiają tworzenie projektów mających realizować określone zadania. Daje się jednak zauważyć, że moduły przeznaczone do posługiwania się danymi fotogrametrycznymi i satelitarnymi są często odrębnymi i kosztownymi pakietami.

W Akademii Rolniczo Technicznej w Olsztynie, w Instytucie Gospodarki Przestrzennej jest rozwijane autorskie oprogramowanie, którego zadaniem jest wspomaganie gospodarowania przestrzenią. Główny nacisk jest położony na wykorzystanie danych fotogrametrycznych oraz odniesienie tych danych do przestrzennego aspektu własności gruntów.

Środowiskiem programistycznym jakie zostało wybrane do realizacji zadania jest pakiet Builder C++ firmy Borland. System operacyjny to Windows NT 4.0 WorkStation. W skład pakietu wchodzi Borland Database Engine. Jest to środowisko relacyjnej bazy

danych, w którym z poziomu kodu źródłowego można tworzyć tabele, dodawać, usuwać i aktualizować rekordy oraz przy pomocy składni SQL realizować algebrę relacji, której podstawowymi operatorami są selekcja, rzut i złączenie.

2. Architektura programu

Aplikacja została zaprojektowana według architektury Multi Document Interfejs (MDI). Zaletą takiego podejścia jest jednorazowe zdefiniowanie klasy MDIChild (okna potomnego, w którym wyświetlany jest obraz), wraz ze wszystkimi jej metodami, przy czym w jednym czasie może być wywołanych wiele instancji tej klasy.

Obraz może być wyświetlany na trzy różne sposoby:

- wyświetlanie pojedynczego obrazu;
- wyświetlanie sumy dwóch różnych obrazów (anaglify);
- wyświetlanie dwóch obrazów obok siebie celem obserwacji stereoskopowej.

Obraz wyświetlany jest w kontekście urządzenia HDC komponentu PaintBox, który wywodzi się z klasy GraphicObject i jest obiektem klasy MDIChild.

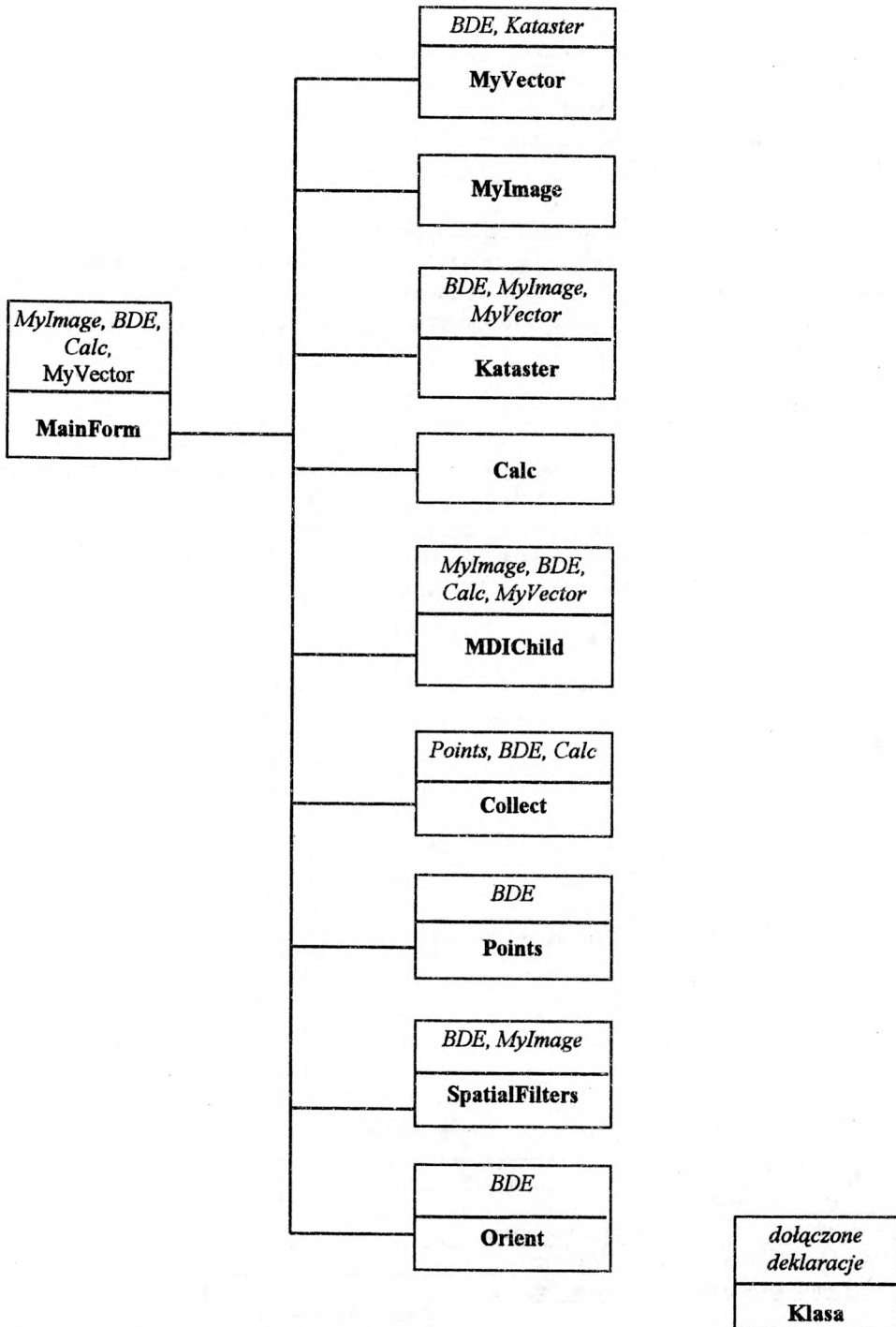
Celem przyspieszenia obsługi zdarzenia OnPaint, czyli odświeżania obrazu na monitorze zadeklarowano trzy różne komponenty PaintBox: dla pojedynczego obrazu, dla anaglifów i dla pary obrazów. Każdy z nich ma zdefiniowaną inną funkcję PaintIt wywoływaną przez OnPaint. W chwili wywoływania konstruktora klasy MDIChild jeden z komponentów PaintBox określany jest jako aktywny. W ten sposób uniknięto konieczności analizy, który z obrazów ma być wyświetlany przy obsłudze zdarzenia OnPaint.

Grafika wektorowa jest obsługiwana przy pomocy struktur potrzebnych do zapisu pliku oraz prezentacji obiektów graficznych w pamięci komputera.

Część opisowa ewidencji gruntów została ujęta w postaci tabel bazy danych.

Na poniższym schemacie pokazano ważniejsze klasy aplikacji:

- MainForm - główne okno aplikacji;
- MDIChild - okna potomne służące do wyświetlania obrazów;
- MyVector - klasa zaprojektowana do obsługi obrazu wektorowego;
- Kataster - klasa zapewniająca import i dostęp do danych opisowych ewidencji gruntów;
- MyImage - klasa zaprojektowana do obsługi obrazu rastrowego;
- Calc - obliczanie parametrów transformacji, wyrównanie metodą najmniejszych kwadratów;
- Collect - okno do kolekcji par punktów przy transformacjach Helmerta, afinicznej i DLT;
- Points - okno wyboru z bazy punktów;
- SpatialFilters - okno do konstrukcji i wyboru filtrów przestrzennych;
- Orient - procedury wyznaczania elementów orientacji wzajemnej i bezwzględnej.



Rys. 1: Schemat architektury programu

3. Część opisowa ewidencji gruntów

Część opisowa ewidencji gruntów została ujęta w trzech tabelach udostępnianych przez BDE:

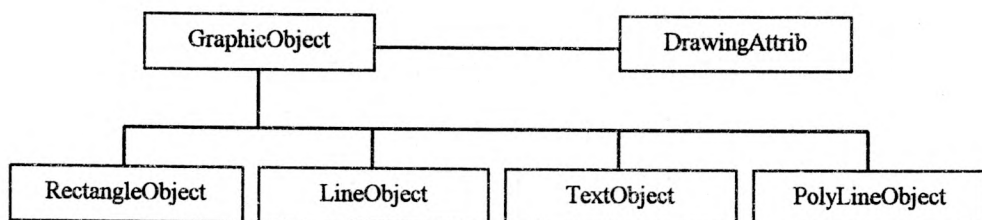
- skorowidz działek ewidencyjnych,
- skorowidz numerów jednostek rejestrowych,
- skorowidz władających.

Schematy formalne tabel mają następującą postać [Kosakowski J., 1997]:

Działki (nr_jednostki_rejestrowej, nr_działki, powierzchnia_działki);
 Jednostki_rejestrowe (nr_jednostki_rejestrowej);
 Władający (nr_jednostki_rejestrowej, nazwa, forma_władania, adres, ...).

4. Obraz wektorowy

Prezentację wektorową w programie mają dane dotyczące działek ewidencyjnych i planu przestrzennego zagospodarowania (granice, numery). Została zaprojektowana klasa MyVector do obsługi obrazu wektorowego. Hierarchia obiektów w tej klasie oraz struktury potrzebne do zapisu pliku i prezentacji obiektów graficznych zostały przyjęte według Nabayjoti Barkakati z drobnymi zmianami. [Barkakati N., 1993].



Rys. 2: Hierarchia obiektów graficznych w klasie MyVector

Format pliku.

4-bajtowa struktura 'IMDG'
60 bajtów komentarza
szerokość pola roboczego (short int)
wysokość pola roboczego (short int)
56 bajtów (niewykorzystane)
4-bajtowy adres pierwszego elementu graficznego
2-bajtowy identyfikator elementu graficznego
4-bajtowy adres następnego elementu graficznego

cd. tabeli na następnej stronie

blok danych elementu graficznego	
2-bajtowy identyfikator elementu graficznego	
4-bajtowy adres następnego elementu graficznego	
blok danych elementu graficznego	
2-bajtowy identyfikator elementu graficznego	
4-bajtowy adres (0 = ostatni element)	
blok danych elementu graficznego	

Struktura bloku danych elementu graficznego zależy od typu obiektu określonego przez dwubajtowy identyfikator tego elementu. Poniżej pokazano struktury Rectangle oraz PolyLine.

Priorytet (short)
współrzędna x lewego górnego rogu (short)
współrzędna y lewego górnego rogu (short)
współrzędna x prawego dolnego rogu (short)
współrzędna y prawego dolnego rogu (short)
struktura DrawingAttributes

Priorytet (short)	
Liczba punktów (short)	
współrzędna x pierwszego punktu (short)	
współrzędna y pierwszego punktu (short)	
współrzędna x drugiego punktu (short)	
współrzędna y drugiego punktu (short)	
blok danych elementu graficznego	
blok danych elementu graficznego	
współrzędna x ostatniego punktu (short)	
współrzędna y ostatniego punktu (short)	
struktura DrawingAttributes	

Struktura DrawingAttributes została nieznacznie zmieniona.

numer warstwy (1 bajt)
link (2 bajty)

Parametry warstw (typ, grubość i kolor linii) mogą być zapisane w oddzielnym pliku lub tabeli.

Pole link posłuży do zapisu numeru połączenia elementu graficznego z rekordem w tabeli bazy danych.

5. Obraz rastrowy

Obraz rastrowy program przechowuje w pamięci jako DIB (Device Independent Bitmap - bitmapa niezależna od urządzenia). Struktura tej bitmapy jest taka sama jak pliku graficznego w formacie BMP, dlatego zdecydowano się na czytanie i zapis obrazów w tym formacie [Barkakati N., 1993; Russ J., 1995].

Zarys struktury BMP.

BitmapFileHeader
BitmapInfoHeader
Paleta kolorów (tablica struktur RGBQUAD)
Tablica pikseli

Zarys struktury DIB.

BitmapInfoHeader
Paleta kolorów (tablica struktur RGBQUAD)
Tablica pikseli

```
typedef struct tagBITMAPFILEHEADER {
    WORD    bfType;
    DWORD   bfSize;
    WORD    bfReserved1;
    WORD    bfReserved2;
    DWORD   bfOffBits;
} BITMAPFILEHEADER;
```

```
typedef struct tagBITMAPINFOHEADER{
    DWORD   biSize;
    LONG    biWidth;
    LONG    biHeight;
    WORD    biPlanes;
    WORD    biBitCount
    DWORD   biCompression;
    DWORD   biSizeImage;
    LONG    biXPelsPerMeter;
    LONG    biYPelsPerMeter;
    DWORD   biClrUsed;
    DWORD   biClrImportant;
} BITMAPINFOHEADER;
```

```
typedef struct tagRGBQUAD {
    BYTE    rgbBlue;
    BYTE    rgbGreen;
    BYTE    rgbRed;
    BYTE    rgbReserved;
} RGBQUAD;
```

Do czytania i zapisu pliku oraz operacji na obrazie została zaimplementowana klasa `MyImage`. Klasa ta nie jest związana z żadnym obiektem graficznym, natomiast jej deklaracja jest dołączona w plikach nagłówkowych wszystkich klas, które mają dostęp do danych obrazu.

W klasie `MyImage` zadeklarowano między innymi następujące funkcje:

- `MyImage :: read (...)` - czysta funkcja wirtualna;
- `MyImage :: write (...)` - czysta funkcja wirtualna;
- `MyImage :: readImage (...)` - odczyt pliku BMP;
- `MyImage :: readPartOfImage (...)` - odczyt dowolnego fragmentu pliku BMP;
- `MyImage :: writeImage (...)` - zapis pliku BMP;
- `MyImage :: makeHist (...)` - budowa histogramu obrazu;
- `MyImage :: drawHist (...)` - rysowanie histogramu w zadanym kontekście urządzenia;
- `MyImage :: showDib (...)` - wyświetlanie Dib w zadanym kontekście urządzenia bez konieczności zamiany na DDB (device dependent bitmap);
- `MyImage :: stretchDib (...)` - powiększanie i pomniejszanie Dib w zadanym kontekście urządzenia bez konieczności zamiany na DDB;
- `MyImage :: filterIt (...)` - filtracja obrazu w dziedzinie przestrzennej metodą splotu;
- `MyImage :: getPixelValue (...)` - czytanie wartości piksela o zadanych współrzędnych obrazu;
- `MyImage :: makeImage (...)` - tworzenie pustej Dib z alokacją pamięci według zadanych parametrów (wysokość, szerokość, liczba bitów na piksel).

6. Plakat posterowy

Na plakacie posterowym przedstawiono:

- 1). tabele zawierające dane opisowe ewidencji gruntów;
- 2). obraz anaglifowy;
- 3). obraz numerycznego modelu terenu uzyskany przez wektoryzację warstwic na mapie topograficznej;
- 4). obraz przekształcenia rzutowego przestrzeni na płaszczyznę.

Przekształcenie zostało wykonane według wzorów [Sitek Z. 1991]:

$$x_i = \frac{AX + BY + CZ + D}{EX + FY + GZ + 1} \quad y_i = \frac{HX + IY + JZ + K}{EX + FY + GZ + 1}$$

Obraz docelowy był próbkowany z obrazu źródłowego metodą najbliższego sąsiedztwa Russ J. [1995].

Współrzędna Z została uzyskana z numerycznego modelu terenu.

Współrzędne x_i , y_i są w układzie znaczków tłowych po uprzedniej transformacji Helmera z układu obrazu.

8. Zakończenie

Środowisko kompilatora C++ Builder firmy Borland okazało się dość przyjaznym i umożliwiającym sprawne oprogramowanie algorytmu grafiki wektorowej i rastrowej. Borland Database Desktop ma słabo rozwinięty interfejs w porównaniu na przykład z MS Access. Mimo to nie można mieć zastrzeżeń do BDE, ponieważ dysponuje chyba wszystkimi funkcjami, dostępnymi z poziomu kodu źródłowego, potrzebnymi do obsługi nawet skomplikowanych operacji na bazie danych. Zagadnienia związane z zastosowaniem fotogrametrycznych zdjęć lotniczych w projektach GIS są atrakcyjne z naukowo badawczego punktu widzenia, ze względu na coraz większą popularność tych danych w Polsce oraz ciekawe aspekty techniczne.

W opisywanym programie mechanizmy związane z grafiką wektorową oraz orientacją wzajemną i bezwzględną stereogramu są nadal w opracowaniu.

Literatura

1. Barkakati N., 1993, *Imaging and Animation for Windows*, Prentice Hall International, England;
2. Kosakowski J., 1997, *Wykorzystanie środowiska relacyjnej bazy danych jako platformy roboczej do przetwarzania danych przy odnowieniu ewidencji gruntów*, Przegląd Geodezyjny, Sigma, Warszawa;
3. Muraszkiewicz M., 1993, *Bazy danych*, Akademicka Oficyna Wydawnicza RM, Warszawa;
4. Russ J., 1995, *The Image Processing Handbook*, CRC Press, London;
5. Sitek Z., 1991), *Fotogrametria ogólna i inżynierska*, Państwowe Przedsiębiorstwo Wydawnictw Kartograficznych, Warszawa, Wrocław;
6. Chou Y., 1997, *Exploring Spatial Analysis in Geographics Information Systems*, ONWORD PRESS, Santa Fe.

Recenzowała: dr inż. Beata Hejmanowska