7. Smith David, Post-Relational Database:Revitalizing Relational Technology for New Applications. - IDC White Papper, March 1994.
8. ANSI X3.135-1992, American National Standard for Information Systems - Database Language - SQL, November, 1992.
9. ANSI/X3/SPARC DMBS Framework. Report of the study group on data base management systems Inform. Systems, 1978

ПРОГРАММНЫЙ МОДУЛЬ ДЛЯ РАБОТЫ С ИЗОБРАЖЕНИЯМИ

Молочко Д. А., Муха В. С.
Белорусский государственный университет информатики и радиоэлектроники
Минск, ул П.Бровки,6, e-mail: mukha@gw.bsuir.unibel.by

В работе описывается разработанный авторами программный модуль для языка BORLAND PASCAL 7.0, предназначенный для работы с графическими файлами формата PCX.

Ключевые слова: анализ и обработка изображений, программные средства.

Задача математической обработки изображений является весьма актуальной, о чем свидетельствуют многочисленные публикации [1-4]. В работе описывается разработанный авторами программный модуль для работы с графическими файлами формата PCX. В сравнении с другими программами по обработке изображений данная разработка обладает следующими особенностями.

1. Предназначена для исследователей, занимающихся разработкой и исследованием *собственных* алгоритмов обработки случайных изображений и полей.
2. Рассчитана на пользователей, *не имеющих глубоких знаний в области форматов графических файлов*, перед которыми стоит задача с помощью общедоступных и простых средств (компилятор языка BORLAND PASCAL 7.0) обрабатывать вводимые изображения и сохранять полученные изображения.
3. Ориентирована на использование *многомерно-матричного подхода* [4] к обработке случайных изображений и полей, когда изображение представляется двумерной матрицей, а его ковариационная матрица – четырехмерной матрицей.

Разработаны и реализованы программно с использованием предложенного модуля новые алгоритмы зашумления изображений и пропускания изображений через систему линз. Модуль и программа могут быть предложены

пользователю, занимающемуся разработкой и исследованием алгоритмов фильтрации и фокусировки изображений и полей.

Программный модуль выполнен в виде модуля языка программирования PASCAL по аналогии с модулем CRT. Он подключается к разрабатываемой программе пользователя командой языка PASCAL uses pcximage;

Разработанный программный модуль предоставляет обьекты для работы с памятью, с графическими файлами формата PCX и просмотра изображения.

Для работы с памятью выделены следующие обьекты :

1. Matrix2Real – обьект для работы с двумерной матрицей, содержащей вещественные числа.

   Этот обьект имеет следующее описание:

```
type Matrix2Real = object
     Values : ^Matrix2r; {указатель на массив указателей, которые указывают на одномерные массивы чисел типа real}
     n, m : integer; {n x m – размерность матрицы}
     procedure Create (n, m : integer); {процедура создания двумерной матрицы размерностью n x m}
     procedure Free; {процедура освобождения памяти, занимаемой двумерной матрицей}
     end;
```

2. Matrix2Byte – обьект для работы с двумерной матрицей, содержащей числа типа byte и имеющий следующее описание:

```
type Matrix2Byte = object
     Values  : ^Matrix2b; {указатель на массив указателей, которые указывают на одномерные массивы чисел типа byte}
     n, m : integer; {n x m – размерность матрицы}
     procedure Create (n, m : integer); {процедура создания двумерной матрицы размерностью n x m}
     procedure Free; {процедура освобождения памяти, занимаемой двумерной матрицей}
     end;
```

3. Matrix4Real - обьект для работы с четырехмерной матрицей, содержащей вещественные числа и имеющий следующее описание:

```
type Matrix4Real = object
```

```
  Values   : ^Matrix4r; {указатель на массив указателей, которые указывают на одномерные массивы указателей, указывающих на ... на одномерные массивы чисел типа real}
  n,m,k,l  : integer; {n x m x k x l – размерность матрицы}
  procedure Create (n, m, k, l : integer); {процедура создания четырехмерной матрицы размерностью n x m x k x l}
  procedure Free; {процедура освобождения памяти, занимаемой четырёхмерной матрицей}
  end;
```

Для того, чтобы использовать данные обьекты, их нужно обьявить в разделе переменных, например:

```
Var MyMatrix2 : Matrix2Byte;
  MyMatrix4 : Matrix4Real;
```

Далее в программе их нужно проинициировать :

```
MyMatrix2.Create (12, k);
MyMatrix4.Create (5, 9, 10, 80);
```

Теперь их можно использовать, например, занести другие значения:

```
For i:=1 to 5 do
 For j:=1 to 9 do
  For k:=1 to 10 do
   For l:=1 to 80 do
    MyMatrix4.Values^[i]^[j]^[k]^[l] := random;
```

Когда массивы не нужны, их нужно удалить из памяти следующим образом:

```
MyMatrix4.Free;
MyMatrix2.Free;
```

После удаления массивов обращаться к ним НЕЛЬЗЯ!

Для работы с графическими файлами формата PCX используется модуль TPCXImage, описание которого приводится ниже.

```
type TPCXImage = object
 Red     : Matrix2Byte; {Красная составляющая.}
 Green   : Matrix2Byte; {Зеленая составляющая.}
 Blue    : Matrix2Byte; {Синяя составляющая.}
 Error   : integer;  {Ошибка.}
 NumOfColor : longint; {Кол-во цветов.}
```

```
maxX, maxY  : integer;  {Размер изображения.}

constructor Create (FileName : string); {Конструктор, считывает с диска файл с именем FileName и разбивает его на цветовые составляющие}
destructor Free;                {Деструктор, освобождает память, занятую цветовыми составляющими}

function SaveToFile (FileName : string; colors : integer) : integer; {Функция сохранения изображения в файл}
function ViewImage : integer; {Функция просмотра изображения}
function GetColor (i,j : integer) : TColor; {Функция получения цвета пиксела с координатами i и j}
procedure SetColor (i,j : integer; color : TColor); { Функция установки цвета пиксела с координатами i и j }
end;
```

Цвет пиксела задаётся следующим образом:

Tcolor = 00RRGGBB, где 00 – два нуля, R, G, B – шестнадцатиричные цифры.

Для создания экземпляра данного класса нужно сделать следующее:

```
Var MyImage : TPCXImage;
Далее в программе
MyImage.Create ("c:\pictures\mypicture.pcx");
If MyImage.Error <> NO_ERRORS then
Begin
 Writeln ('ошибка!!!'); halt (-1);
End;
```

Теперь можно обрабатывать изображение.

Сначала красную составляющую :

```
MyImage.Red.Value^[i]^[j] := ...
```

Потом зеленую :

```
MyImage.Green.Value^[i]^[j] := ...
```

И наконец синюю:

```
MyImage.Blue.Value^[i]^[j] := ...
```

После обработки изображение можно сохранить :

MyImage.SaveToFile ("c:\pictures\new.pcx", colors_16mil);

Если изображение не нужно, его можно удалить из памяти следующим образом :

MyImage.Free;

После удаления изображения обращаться к нему НЕЛЬЗЯ !

Функции SaveToFile и ViewImage возвращают код ошибки, которая может принимать одно из следующих значений :

FILE_NOT_FOUND – Файл не найден

FILE_IS_CREATE – Файл уже существует

CANT_CREATE – Не могу создать файл

IT_IS_NOT_PCX_FILE – Файл не PCX формата

NOT_ENOUGH_MEMORY – Не хватает памяти

UNKNOWN_FORMAT – Неизвестный формат файла

NO_ERRORS – Нет ошибок

FILE_WAS_NOT_LOADED – Изображение не загружено

DEVICE_ERROR – Ошибка устройства

Литература

1. Proceeding of Fifth International Conference PRIP'99 "Pattern Recognition and Information Processing". (18-20 May 1999, Minsk, Republic of Belarus). Volume 1. Minsk-Szczecin, 1999.
2. Прэтт У. Цифровая обработка изображений. Кн. 1. М.:Мир, 1982. - 312 с.
3. Прэтт У. Цифровая обработка изображений. Кн. 2. М.:Мир, 1982. - 792 с.
4. Муха В.С. Многомерно-матричная технология в теории моделирования изображений/ Материалы международной научно-технической конференции "Новые информационные технологии в науке и производстве".- Мн.: Белорусский государственный университет информатики и радиоэлектроники, 1998. - С. 199-202.