(cc) BY

Каменева Анастасия Евгеньевна
магистрант
Горбунова Алина Викторовна
студентка
Артамонов Александр Андреевич
магистрант
ФГБОУ ВПО «Магнитогорский государственный
технический университет им. Г.И. Носова»
г. Магнитогорск, Челябинская область

ПРОСТРАНСТВЕННАЯ ФИЛЬТРАЦИЯ ИЗОБРАЖЕНИЯ НА ПРИМЕРЕ СГЛАЖИВАЮЩЕГО ФИЛЬТРА

Аннотация*: в статье рассматривается реализация пространственных фильтров на языке C# и сравниваются фильтры Гаусса различной размерности.*

Ключевые слова*: обработка изображений, алгоритмы, матрицы.*

При решении задач обработки и анализа изображений используют следующую дискретную модель цифрового изображения [2]:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N\text{-}1) \\ \vdots & \cdots & \cdots & \vdots \\ f(M\text{-}1,0) & f(M\text{-}1,1) & \cdots & f(M\text{-}1,N\text{-}1) \end{bmatrix} \quad (1)$$

Окрестностью вокруг точки (x, y) называется квадратная или прямоугольная область с центром в точке (x, y).

Фильтрация изображения f, имеющего размеры M x N, с помощью фильтра размерами m x n задается выражением общего вида:

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t), \quad (2)$$

Процедура фильтрации изображения сводится к умножению коэффициентов матрицы маски на элементы матрицы, включающей в себя пиксели изображения, захваченные окном [1].

Рассмотрим реализацию сглаживающего фильтра с масками разного размера A_1 (3x3) и A_2(5x5) (размытие по Гауссу) на языке C# и сравним результаты.

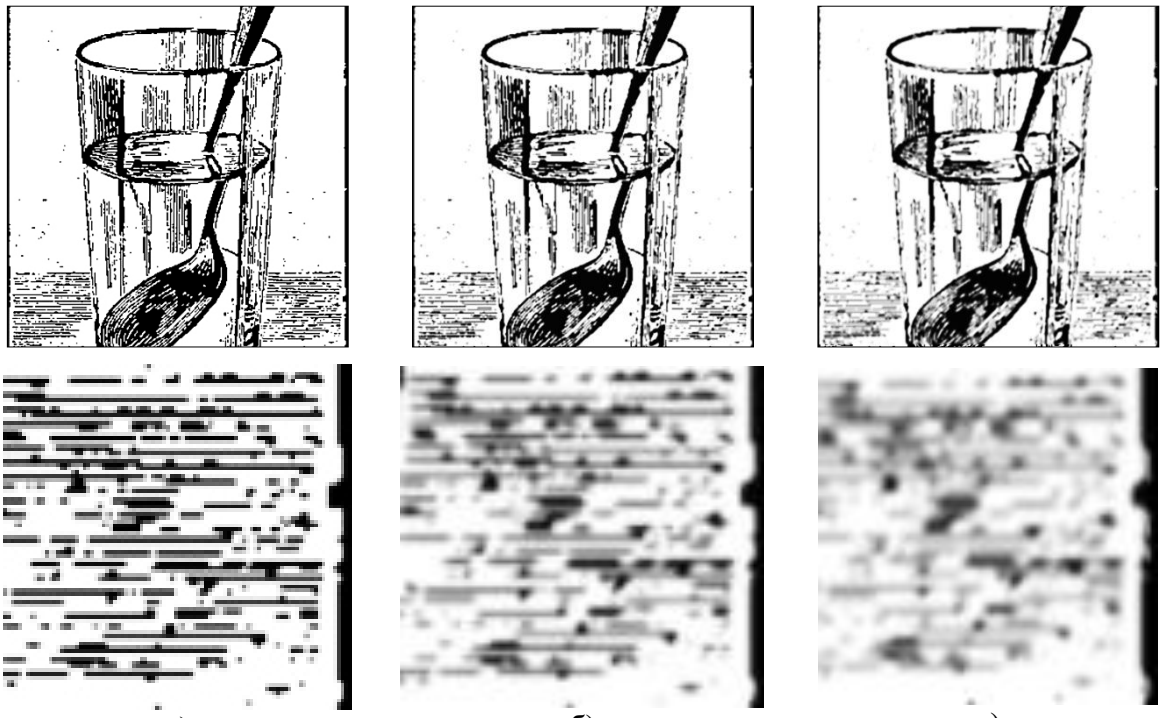
```
WriteableBitmap Filter(WriteableBitmap originalImage, double[,] convMatr, double div, double size) {
Color[,] kernel = new Color[size, size];
byte A;
int sumR, sumG, sumB;
WriteableBitmap wb = originalImage;
for (int x = 0; x <= wb.PixelWidth – size; x++) {
for (int y = 0; y <= wb.PixelHeight – size; y++) {
A = wb.GetPixel(x + 1, y + 1).A;
sumR = sumG = sumB = 0;
for (int i = 0; i < size; i++) {
for (int j = 0; j < size; j++)
kernel[i, j] = wb.GetPixel(x + i, y + j);
}
for (int i = 0; i < size; i++) {
for (int j = 0; j < size; j++) {
sumR += (int)(kernel[i, j].R * convMatr[j, i] / div);
sumG += (int)(kernel[i, j].G * convMatr[j, i] / div);
sumB += (int)(kernel[i, j].B * convMatr[j, i] / div);
}
}
Color newCl = new Color();
newCl.A = A;
newCl.R = sumR < 0? 0 : (sumR>255?255: sumR);
newCl.G = sumG < 0? 0 : (sumG>255?255: sumG);
newCl.B = sumB < 0? 0 : (sumB>255?255: sumB);
wb.SetPixel(x + 1, y + 1, newCl);
}
}
return wb;
```

}

Сравним результаты применения процедуры Filter с различными сглаживающими фильтрами:

$$A_1 = \frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}, A_2 = \frac{1}{159}\begin{pmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{pmatrix}.$$

Заметим, что маска A_2 показала более сильное размытие, чем маска A_1. Однако, при работе программы маска A_2 демонстрирует меньшую вычислительную эффективность – из-за бóльшего числа арифметических операций реализация фильтра A_2 работала дольше, чем A_1. Особенно заметна разница во времени обработки для изображений большого разрешения.



а) б) в)

Рис. 1. Результаты работы функции Filter: а) исходное изображение;

б) результат обработки изображения маской A_1;

в) результат обработки изображения маской A_2

Список литературы

1. Каменева А.Е. Средства Windows Phone 8.1 Silverlight для создания модуля обработки изображений / А.Е. Каменева, А.В. Горбунова // Научное сообщество студентов XXI столетия: Сб. ст. по мат. XLII междунар. студ. науч.-практ. конф. – №5(41).

2. Каменева А.Е. Градационные преобразования матрицы изображения [Текст] / А.Е. Каменева, А.В. Горбунова // Электронный научный журнал: Сб. статей. – М., 2016. – №3 (3).