МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра ТВ

ОТЧЁТ

по лабораторной работе № 1

по дисциплине «Цифровая обработка изображений»

Тема: РАЗРАБОТКА ПРОГРАММЫ ДЛЯ КВАНТОВАНИЯ ИЗОБРАЖЕНИЙ

Студенты гр. 9105 _____ Шаривзянов Д. Р.

_____ Басманов А. А.

Преподаватель _____ Поздеев А. А.

Санкт-Петербург

2024

КВАНТОВАНИЕ ИЗОБРАЖЕНИЙ

Цели работы: знакомство с гистограммой и методом квантования изображений.

Теоретические сведения

Формула среднеквадратической ошибки квантования:

$$\sigma_e = \sqrt{\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(I(i,j)-K(i,j))^2},$$

где *I(i, j)* – значение яркости пикселя с координатами (i, j) в исходном полутоновом изображении (с числом уровней квантования 256), *K(i, j)* – значение яркости пикселя с координатами (i, j) в квантованном изображении с меньшим числом уровней квантования (q_level), *m* и *n* – число строк и столбцов изображения.

Оценку среднеквадратической ошибки квантования можно получить, используя значение величины интервала между уровнями квантования:

$$\hat{\sigma}_e = \text{inter} / \sqrt{12}.$$

1. Код программы

```
Mat getHist(const Mat &image)
{
	int hist_h = 400, hist_w = 256*3;

	Mat hist = Mat::zeros(1, 256, CV_64FC1);

	for (int i = 0; i < image.cols; i++)
		for (int j = 0; j < image.rows; j++) {
			int r = image.at<unsigned char>(j, i);
			hist.at<double>(0, r) = hist.at<double>(0, r) + 1.0;
		}
	double m = 0, M = 0;
	minMaxLoc(hist, &m, &M);
	hist = hist / M;
	Mat hist_img = Mat::zeros(100, 256, CV_8U);
	for (int i = 0; i < 256; i++)
		for (int j = 0; j < 100; j++) {
			if (hist.at<double>(0, i) * 100 > j) {
				hist_img.at<unsigned char>(99 - j, i) = 255;
			}
		}
```

```
        bitwise_not(hist_img, hist_img);
        resize(hist_img, hist_img, Size(hist_w, hist_h), 0 , 0, INTER_NEAREST);
        return hist_img;
}
```

```
Mat getQuant(const Mat &image, int q_level)
{
        double sko = 0.0;
        int inter = 255 / (q_level - 1);

        Mat img_quant = Mat::zeros(image.rows, image.cols, CV_8UC1);

        for (int row = 0; row < image.rows; row++){
                for (int col = 0; col < image.cols; col++){

        int Y = image.at<uchar>(row, col);

                        for (int k = 0; k < q_level; k++){
                                if ((Y > inter * k) && (Y <= inter * k + inter / 2))                Y = inter * k;
                                if ((Y > inter * k + inter / 2) && (Y <= inter * (k + 1)))  Y = inter * (k + 1);
                        }
        img_quant.at<uchar>(row, col) = Y;
        sko += (image.at<uchar>(row, col) - img_quant.at<uchar>(row, col)) * (image.at<uchar>(row, col) -
img_quant.at<uchar>(row, col));
                }
        }
   sko /= (image.rows*image.cols);
        sko = sqrt(sko);

        cout << "\tsko = " << sko << " \t" << "ass sko = " << inter / sqrt(12) << endl;
        return img_quant;
}
```

```
void lab1(const Mat &img_bgr){
   Mat img_gray;
   cvtColor(img_bgr, img_gray, COLOR_BGR2GRAY);

   imshow("image bgr", img_bgr);
   imshow("image gray", img_gray);

   cout << "for origin: ";
   Mat quant = getQuant(img_gray, 256);
   imshow("Quantization with " + to_string(256) + " levels", quant);
   Mat hist = getHist(img_gray);
   imshow("histogram origin", hist);

   for (int q = 2; q < 65; q*=2) {
   cout << "for q = " << q << ": ";
   Mat quant = getQuant(img_gray, q);
   imshow("Quantization with " + to_string(q) + " levels", quant);

   Mat hist = getHist(quant);
   imshow("histogram with " + to_string(q) + " levels", hist);
   }

   waitKey();
}
```

2. Исходное изображение и его гистограмма

Рис 1. Исходное полутоновое изображение и его гистограмма.

Значение среднеквадратического отклонения отчетов исходного полутонового изображения равно 0, так как квантования и ошибок не происходит (по формуле ошибки получается вычитание значения исходного пиксела от самого себя).

3. Изображения, квантованные по 2, 4, 8, 16, 32 и 64 уровням, а также их гистограммы.

Рис. 2. Квантование по 2 уровням и гистограмма

Рис. 3. Квантование по 4 уровням и гистограмма

Рис. 4. Квантование по 8 уровням и гистограмма

Рис. 5. Квантование по 16 уровням и гистограмма

Рис. 6. Квантование по 32 уровням и гистограмма

Рис. 7. Квантование по 32 уровням и гистограмма

4. **Таблица с полученными значениями среднеквадратической ошибки квантования.**

q_level	2	4	8	16	32	64	256
σ_e	100.163	26.6323	12.1912	5.04383	2.4747	1.23085	0
$\hat{\sigma}_e$	73.6122	24.5374	10.3923	4.90748	2.3094	1.1547	0.288675

Выводы: в ходе выполнения программы из цветного изображения получается полутоновое путем преобразования значений RGB к яркостному значению с последующим его квантованием. Квантование представляет собой процесс разделения диапазона возможных значений яркости на фиксированное количество уровней, что позволяет уменьшить количество информации, необходимое для представления изображения, с возможностью сохранить при этом его визуальное качество на приемлемом уровне.

Гистограмма распределения уровней яркости визуализирует, как часто каждый уровень яркости встречается в изображении, что является полезным инструментом для анализа изображений, в частности, для оценки контрастности и распределения яркости.

Среднеквадратическое отклонение и среднеквадратическая ошибка квантования позволяют количественно оценить потерю качества изображения в результате его квантования. Эти метрики важны для понимания компромисса между сокращением объема данных изображения и сохранением его качества.

Из результатов лабораторной работы следует, что **с увеличением числа уровней квантования** повышается качество воспроизведения изображения, также **уменьшается значение среднеквадратической ошибки** квантования и ведет к увеличению объема данных. В свою очередь, уменьшение числа уровней квантования сокращает объем данных, но может приводить к ухудшению визуального качества изображения, в том числе к появлению ложных контуров и потере деталей.

Отличие фактического значения ошибки квантования от оценки по величине интервала для используемого изображения незначительно, однако в случае высокодетализированного изображения (рис.8) при некоторых случаях квантования данные значения могут отличаться в несколько раз. Кажется, это ошибка в коде.

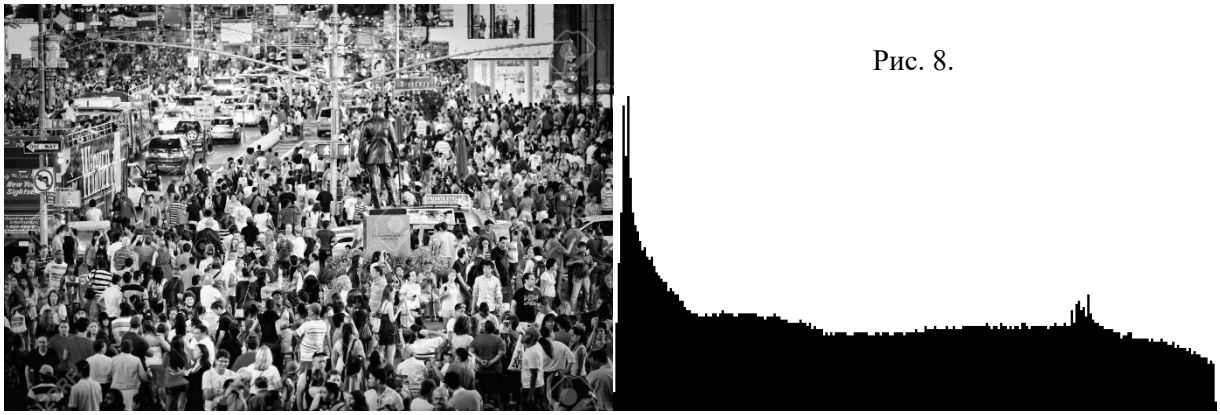

Рис. 8.

Результаты расчёта ошибок квантования для высокодетализированного изображения:

for origin:	sko = 0	ass sko = 0.288675
for q = 2:	sko = 70.1599	ass sko = 73.6122
for q = 4:	sko = 24.0162	ass sko = 24.5374
for q = 8:	sko = 10.4034	ass sko = 10.3923
for q = 16:	sko = 4.94135	ass sko = 4.90748
for q = 32:	**sko = 17.2642**	**ass sko = 2.3094**
for q = 64:	**sko = 5.27716**	**ass sko = 1.1547**
for q = 128:	sko = 0.704901	ass sko = 0.57735