Основные определения:

**Ранговый блок** – подобласть исходного изображения, представляющая из себя квадрат; ранговые блоки не могут пересекаться.

**Доменный блок (домен)** – подобласть исходного изображения, представляющая из себя квадрат со стороной в 2 раза больше стороны рангового блока; доменные блоки могут пересекаться.

**Коэффициент компрессии ε** (коэффициент «похожести») – коэффициент, позволяющий оценить степень схожести рангового и доменного блока.

**k** – степень разбиения рангового блока (изначально k = 1, в дальнейшем значение меняется в зависимости от алгоритма).

**Контрастность *s*** и **Яркость *o*** – яркостные характеристики преобразования доменного блока к ранговому блоку.

Оптимальные контрастность и яркость минимизируют выражение , в котором и это соответственно значения пискелей ранговой и доменной областей, и вычисляются по формулам

Где

В которых N – размер стороны рангового (доменного) блока

В качестве параметров соответствующего доменного блока сохраняют:

* Координату Х верхнего левого угла доменного блока;
* Координату Y верхнего левого угла доменного блока;
* Номер аффинного преобразования доменного блока;
* Степень разбиения рангового блока;
* Контрастность;
* Яркость.

**Общий алгоритм фрактального сжатия**

1. Исходное изображение разбивается на ранговые блоки;
2. Для каждого рангового блока:
   1. Ищется доменный блок, соответствующий ранговому блоку;
   2. Сохраняются параметры подошедшего доменного блока;
3. Полученные соответствия между ранговыми и доменными блоками (фрактальный код) сохраняются для дальнейшего использования.

Подходящий доменный блок может выбираться несколькими способами:

1. До первого найденного доменного блока, удовлетворяющего условию (1). Если ни один доменный блок не удовлетворяет условию (1):
   1. Берем доменный блок с минимальным СКО (алгоритм А1, представлен на рисунке 3);
   2. Разбиваем ранговый блок на 4 блока и для каждого из них ищем подходящий доменный блок (алгоритм А2, представлен на рисунке 4).
2. Доменный блок с минимальным СКО (алгоритм Б, представлен на рисунке 5).

**Алгоритмы поиска подходящего доменного блока.**

**Алгоритм А1**

Входные параметры: исходное изображение, ранговый блок, коэффициент компрессии ε.

Используемые параметры: *minSKO* (значение СКО, соответствующее минимальному СКО из всех, рассчитанных для заданного рангового блока), *minХ* (координата Х верхнего левого угла доменного блока, соответствующего *minSKO*), *minY* (координата Y верхнего левого угла доменного блока, соответствующего *minSKO*), *minAfin* (номер аффинного преобразования доменного блока, соответствующего *minSKO*).

1. Задаем значение исходных данных;
2. Задаем начальные значения для *minSKO*, *minХ*, *minY*, *minAfin*;
3. На исходном изображении выделяем непроверенный доменный блок;
4. Уменьшаем его в 2 раза;
5. Рассчитываем минимальное СКО (*min*) между ранговым блоком и аффинными преобразованиями доменного блока;
6. Если минимальное СКО меньше коэффициента компрессии то сохраняем параметры преобразования текущего доменного блока, иначе – переходим к п.7;
7. Если найденное минимальное СКО меньше значения входного параметра *minSKO* то переходим к п.8, иначе – к п.9;
8. В параметр *minSKO* присваиваем значение *min*, в *minХ*, *minY*, *minAfin* сохраняем соответствующие параметры доменного блока;
9. Если на исходном изображении остались непроверенные доменные блоки, то переходим в п.2, иначе – в п.10;
10. Сохраняем параметры преобразования доменного блока, соответствующего *minSKO*.

**Алгоритм А2**

Входные параметры: исходное изображение, ранговый блок, коэффициент компрессии ε.

Используемые параметры: *minSKO* (значение СКО, соответствующее минимальному СКО из всех, рассчитанных для заданного рангового блока), *minХ* (координата Х верхнего левого угла доменного блока, соответствующего *minSKO*), *minY* (координата Y верхнего левого угла доменного блока, соответствующего *minSKO*), *minAfin* (номер аффинного преобразования доменного блока, соответствующего *minSKO*).

1. Задаем значение исходных данных
2. Задаем начальные значения для *minSKO*, *minХ*, *minY*, *minAfin*;
3. На исходном изображении выделяем непроверенный доменный блок;
4. Уменьшаем его в 2 раза;
5. Рассчитываем минимальное СКО (*min*) между ранговым блоком и аффинными преобразованиями доменного блока;
6. Если минимальное СКО меньше коэффициента компрессии то сохраняем параметры преобразований текущего доменного блока, иначе – переходим к п.7;
7. Если найденное минимальное СКО меньше значения входного параметра *minSKO* то переходим к п.8, иначе – к п.9;
8. В параметр *minSKO* присваиваем значение *min,* в *minХ*, *minY*, *minAfin* сохраняем соответствующие параметры доменного блока;
9. Если на исходном изображении остались непроверенные доменные блоки, то переходим в пункт 2, иначе – в п.10;
10. Если можем разделить ранговый блок на 4 подблока, то переходим к п.11, иначе – п.12;
11. Делим ранговый блок на 4 подблока и для каждого из них выполняем данный алгоритм;
12. Сохраняем параметры преобразований доменного блока, соответствующего *minSKO*.

**Алгоритм Б**

Входные параметры: исходное изображение, ранговый блок.

Используемые параметры: *minSKO* (значение СКО, соответствующее минимальному СКО из всех, рассчитанных для заданного рангового блока), *minХ* (координата Х верхнего левого угла доменного блока, соответствующего *minSKO*), *minY* (координата Y верхнего левого угла доменного блока, соответствующего *minSKO*), *minAfin* (номер аффинного преобразования доменного блока, соответствующего *minSKO*).

1. Задаем значение исходных данных;
2. Задаем начальные значения для *minSKO*, *minХ*, *minY*, *minAfin*;
3. На исходном изображении выделяем непроверенный доменный блок;
4. Уменьшаем его в 2 раза;
5. Рассчитываем минимальное СКО (*min*) между ранговым блоком и аффинными преобразованиями доменного блока;
6. Если найденное минимальное СКО меньше значения входного параметра *minSKO* то переходим к п.7, иначе – к п.8;
7. В параметр *minSKO* присваиваем значение *min,* в *minХ*, *minY*, *minAfin* сохраняем соответствующие параметры доменного блока;
8. Если на исходном изображении остались непроверенные доменные блоки, то переходим в пункт 2, иначе – в п.9;
9. Сохраняем параметры преобразований доменного блока, соответствующего *minSKO*.

**Алгоритм декомпрессии**

Входные параметры: базовое изображение, k – количество итераций декодирования, фрактальный код.

1. Задаем значение исходных данных;
2. Из фрактального кода выделяем параметры преобразований для рангового блока;
3. На базовом изображении выделяем соответствующий параметрам доменный блок;
4. Уменьшаем его;
5. Применяем аффинное преобразование;
6. Применяем яркостное преобразование;
7. Копируем преобразованный доменный блок на место текущего рангового блока в базовом изображении;
8. Если декодировали все ранговые блоки, то переходим к п.9, иначе – к п.2;
9. Повторяем данный выполнение данного алгоритма k раз

**Предварительная классификация блоков**

В случае использования предварительной классификации блоков, прежде чем приступить к выполнению компрессии, каждому ранговому и доменному блоку, выделяемом на изображении, присваивается определенный класс; в дальнейшем, поиск подходящего доменного блока осуществляется только среди доменных блоков, имеющих тот же класс, что и ранговый блок.

В данной работе для классификации блоков я использую следующие подходы:

1. Классификация по значению центра массы блока;
2. Классификация по разнице граничных яркостных значений блока.

Центра масс блока рассчитывается по формуле 1:

В которой N – количество пикселей в блоке, - яркостная характеристика пикселя.

**Метод эталонного блока**

Входные параметры: исходное изображение, эталонный блок, коэффициент компрессии ε.

Используемые параметры: *domensSKOs[j][s]* (матрица значений среднеквадратических отклонений афинных преобразований доменного блока от эталонного блока, где - номер доменного блока, - номер аффинного преобразования.

1. Задаем значение исходных данных;
2. Рассчитываем СКО между эталонным блоком и каждым аффинным преобразованием всех доменных блоков; сохраняем рассчитанные значения в матрицу *domensSKOs*;
3. Выделяем ранговый блок;
4. Рассчитываем *rangSKO* (значение СКО между эталонным и ранговым блоками);
5. Рассчитываем *min* (минимальный модуль разницы между *rangSKO* и значениями матрицы *domensSKOs*);
6. Если *min* меньше ε, то переходим к п.7, иначе – к п.8;
7. Сохраняем параметры преобразования доменного блока, соответствующего *min*; переходим к п.9;
8. Для текущего доменного блока выполняем алгоритм А1, А2 или Б;
9. Если закодированы не все ранговые блок, то переходим к п.3, иначе – заканчиваем выполнение алгоритма.

**Литература**

1. Кудрина М.А., Климентьев К.Е. Компьютерная графика. − Издательство СГАУ, 2013. – 140 с.
2. Ансон Л., Барнсли М. Фрактальное сжатие изображения //Мир ПК, 1992, № 4, с. 52 – 58.