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. Вычисляется как среднее квадратическое отклонение (СКО):

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

**Контрастность *s*** и **Яркость *o*** – параметры 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 содержит следующие шаги:

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 сохраняются в файл.

Подходящий доменный блок может выбираться несколькими способами:

1. Первый встречный доменный блок, удовлетворяющий условие формулы 1. Если ни один доменный блок не удовлетворяет условию:
   1. Берем доменный блок с минимальный СКО;
   2. Разбиваем ранговый блок на 4 блока и для каждого из них ищем подходящий доменный блок.
2. Доменный блок с минимальным СКО;

Для ускорения процесса сжатия можно выделить 2 подхода:  
1) Предварительная классификация блоков [2];

2) Метод «эталонного» блока.

**Алгоритмы поиска подходящего доменного блока.**

**Первый подходящий доменный блок (без разбиения).**

Входные параметры: исходное изображение, ранговый блок, коэффициент компрессии ε, параметр *minSKO* (значение СКО, соответствующее минимальному СКО из всех, рассчитанных для заданного рангового блока).

1. Задаем значение исходных данных
2. На исходном изображении выделяем непроверенный доменный блок;
3. Уменьшаем его в 2 раза;
4. Рассчитываем минимальное СКО (*min*) между ранговым блоком и аффинными преобразованиями доменного блока;
5. Если минимальное СКО меньше коэффициента компрессии то сохраняем параметры доменного блока, иначе – переходим к п. 6;
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*;
8. Если на исходном изображении остались непроверенные доменные блоки, то переходим в пункт 2, иначе – в п. 9;
9. Сохраняем параметры доменного блока, соответствующего minSKO.

**Первый подходящий доменный блок (с разбиением).**

Входные параметры: исходное изображение, ранговый блок, коэффициент компрессии ε, параметр *minSKO* (значение СКО, соответствующее минимальному СКО из всех, рассчитанных для заданного рангового блока).

1. Задаем значение исходных данных
2. На исходном изображении выделяем непроверенный доменный блок;
3. Уменьшаем его в 2 раза;
4. Рассчитываем минимальное СКО (*min*) между ранговым блоком и аффинными преобразованиями доменного блока;
5. Если минимальное СКО меньше коэффициента компрессии то сохраняем параметры доменного блока, иначе – переходим к п. 6;
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*;
8. Если на исходном изображении остались непроверенные доменные блоки, то переходим в пункт 2, иначе – в п. 9;
9. Сохраняем параметры доменного блока, соответствующего minSKO.

Применяем аффинное преобразование ***i*** к domen

Ищем СКО между rang и domen

min = Min(afinSKO)

rang

ε

afinSKO[8]

minSKO = 10000000

i = 1

Сохраняем СКО в afinSKO[***i***]

i > 8

Да

Нет

i = i + 1

Рисунок 2 − Расчет минимального СКО между ранговым блоком и аффинными преобразованиями доменного блока

Выделяем доменный блок domen

Рассчитываем min (минимальное СКО между ранговым блоком и аффинными преобразованиями доменного блока)

Уменьшаем domen в 2 раза

rang

ε

minSKO = 10000000

minSKO = min

min < ε

Сохранение параметров domen

Проверили все доменные блоки?

Сохранение параметров доменного блока, соответствующего minSKO

Да

Нет

Да

min < minSKO

Нет

Да

Нет

Рисунок 3 − Поиск подходящего доменного блока 1) a

Выделяем доменный блок domen

Рассчитываем min (минимальное СКО между ранговым блоком и аффинными преобразованиями доменного блока)

Уменьшаем domen в 2 раза

rang

ε

minSKO = 10000000

minSKO = min

Проверили все доменные блоки?

Сохранение параметров доменного блока, соответствующего minSKO

Нет

min < minSKO

Да

Нет

Да

Рисунок 4 − Поиск подходящего доменного блока 1) б

Выделяем доменный блок domen

Рассчитываем min (минимальное СКО между ранговым блоком и аффинными преобразованиями доменного блока)

Уменьшаем domen в 2 раза

rang

ε

minSKO = 10000000

minSKO = min

min < ε

Сохранение параметров domen

Проверили все доменные блоки?

Сохранение параметров доменного блока, соответствующего minSKO

Да

Нет

Да

min < minSKO

Нет

Да

Нет

minSKO < ε

Да

Нет

Делим блок rang на 4 подблока

Для каждого подблока ищем подходящий доменный блок

Можем разделить rang на подблоки?

Да

Нет

Рисунок 5 − Поиск подходящего доменного блока 2)

**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.