Сжатие изображений
(введение)

Дмитрий Ватолин

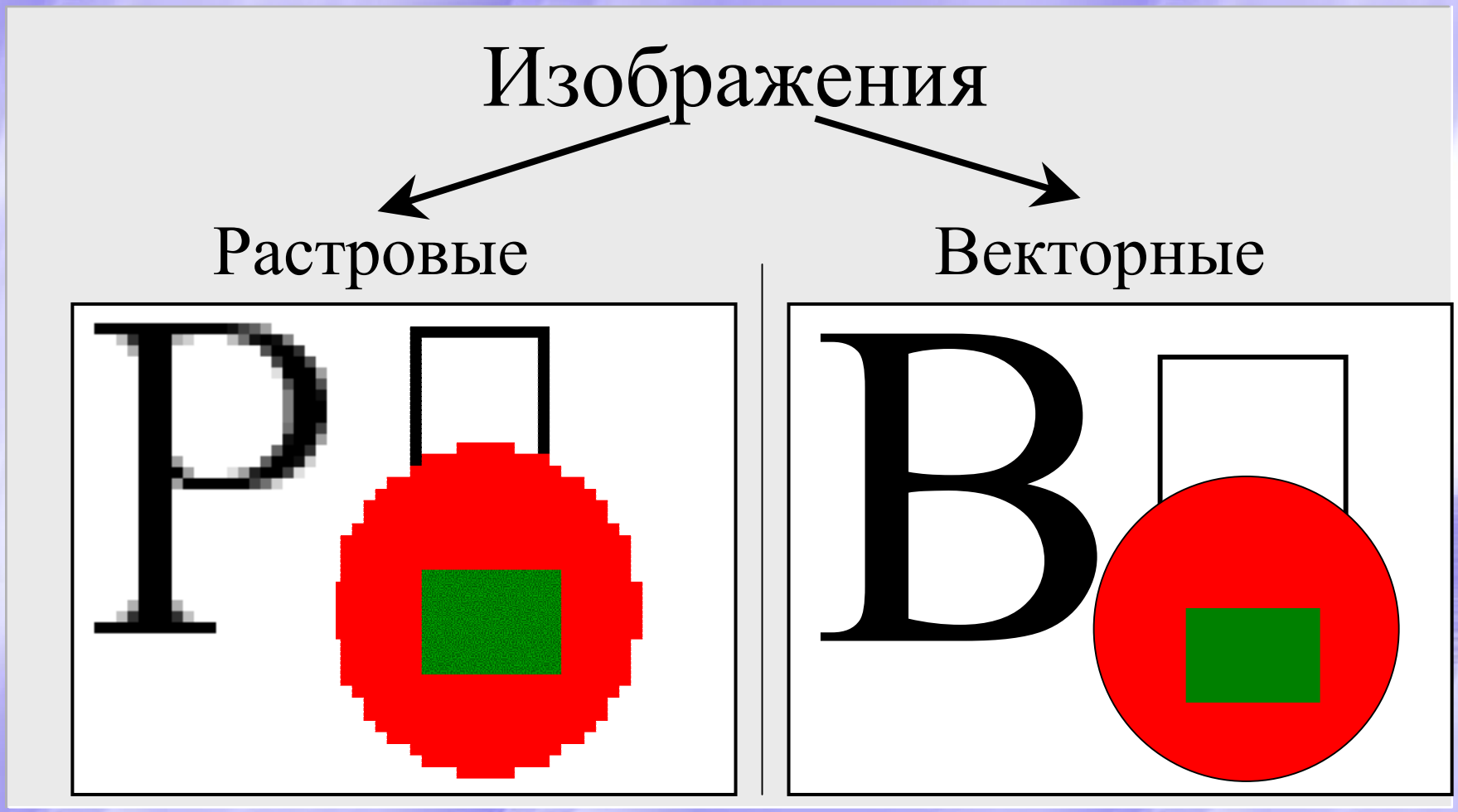
Московский Государственный Университет
CS MSU Graphics&Media Lab

Лекция 14

12 мая 2006

Version 3.3

Типы изображений

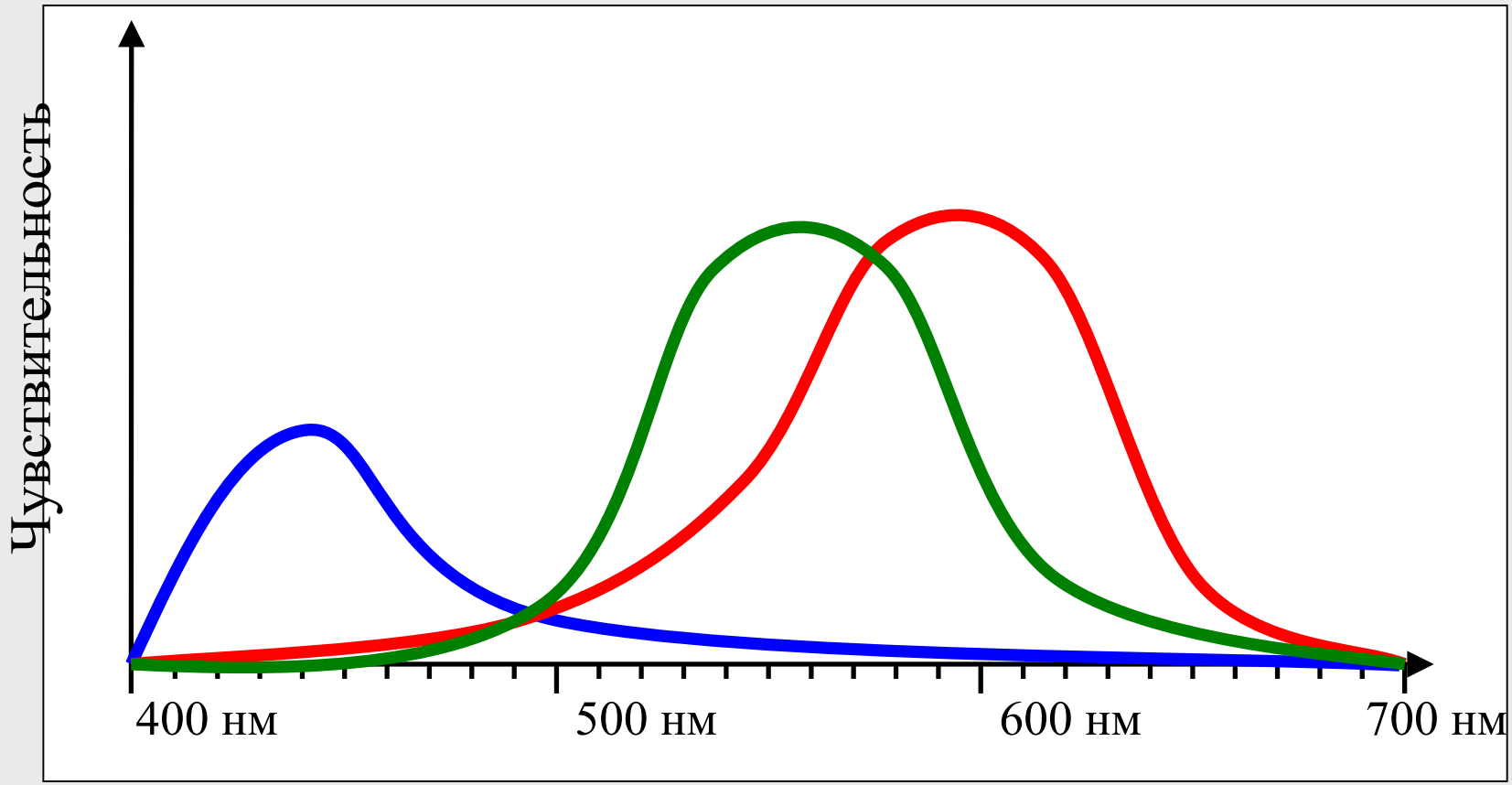


Типы изображений

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Векторные
- Растровые
  - Палитровые
  - Безпалитровые
    - В системе цветопредставления RGB, CMYK, …
    - В градациях серого

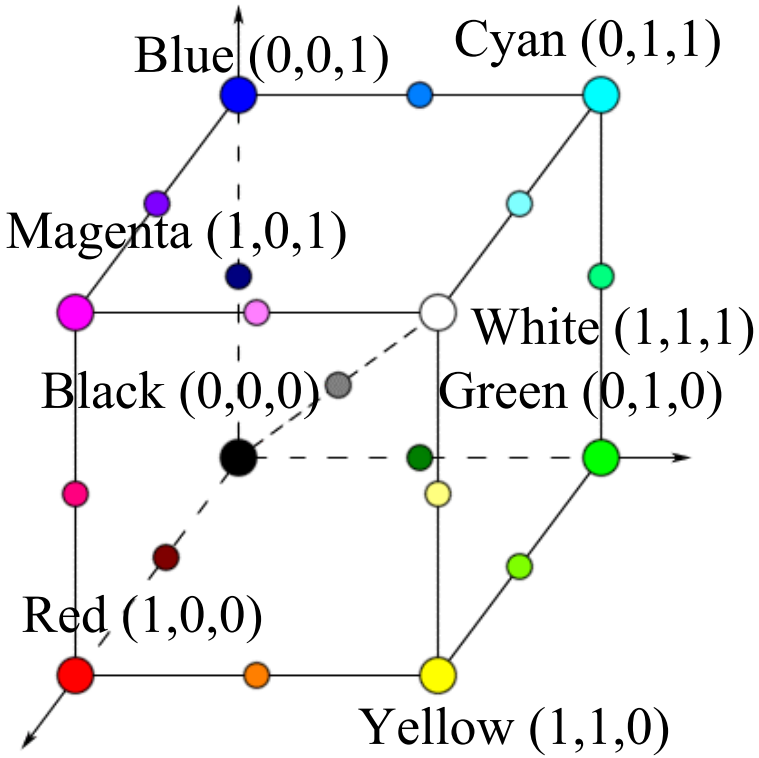
Восприятие цвета

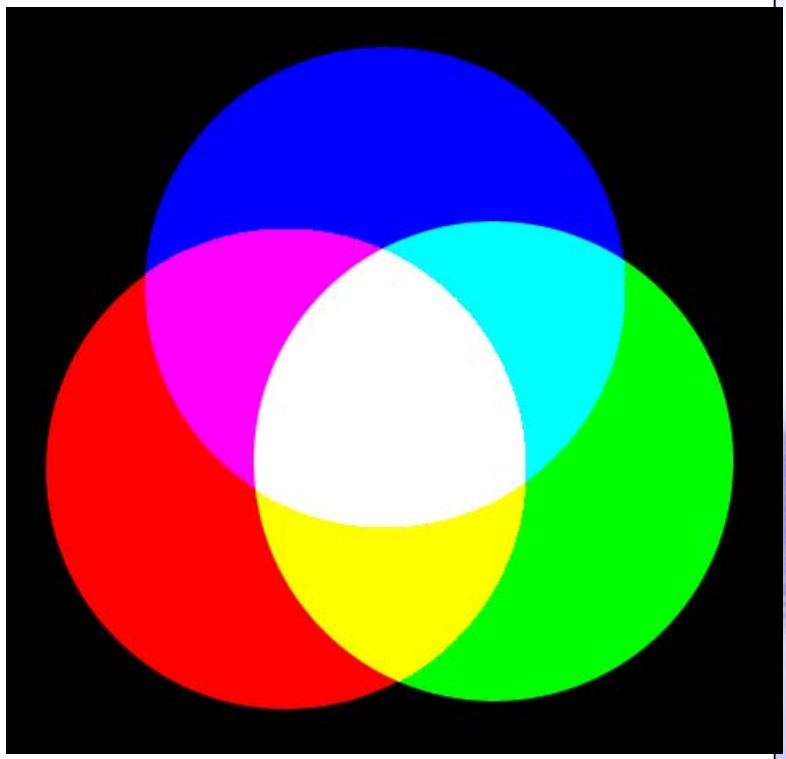


Пространство RGB

GRAPHICS & MEDIA LAB
VIDEO GROUP

RGB (Red, Green, Blue)

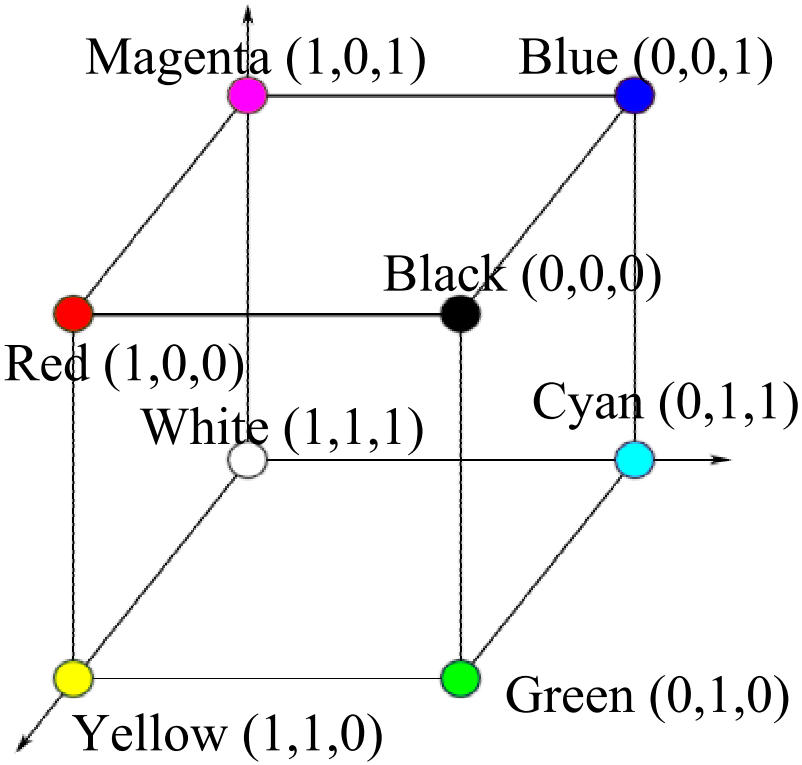


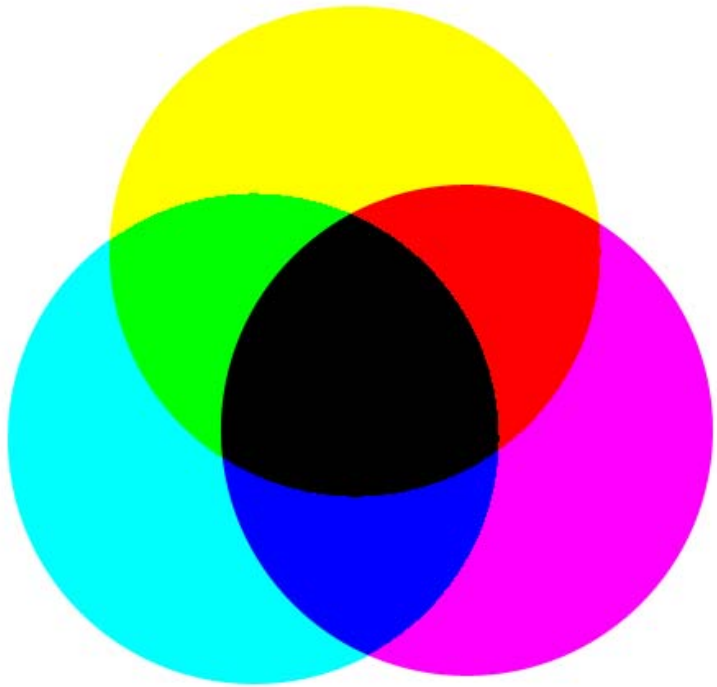


Пространство CMYK

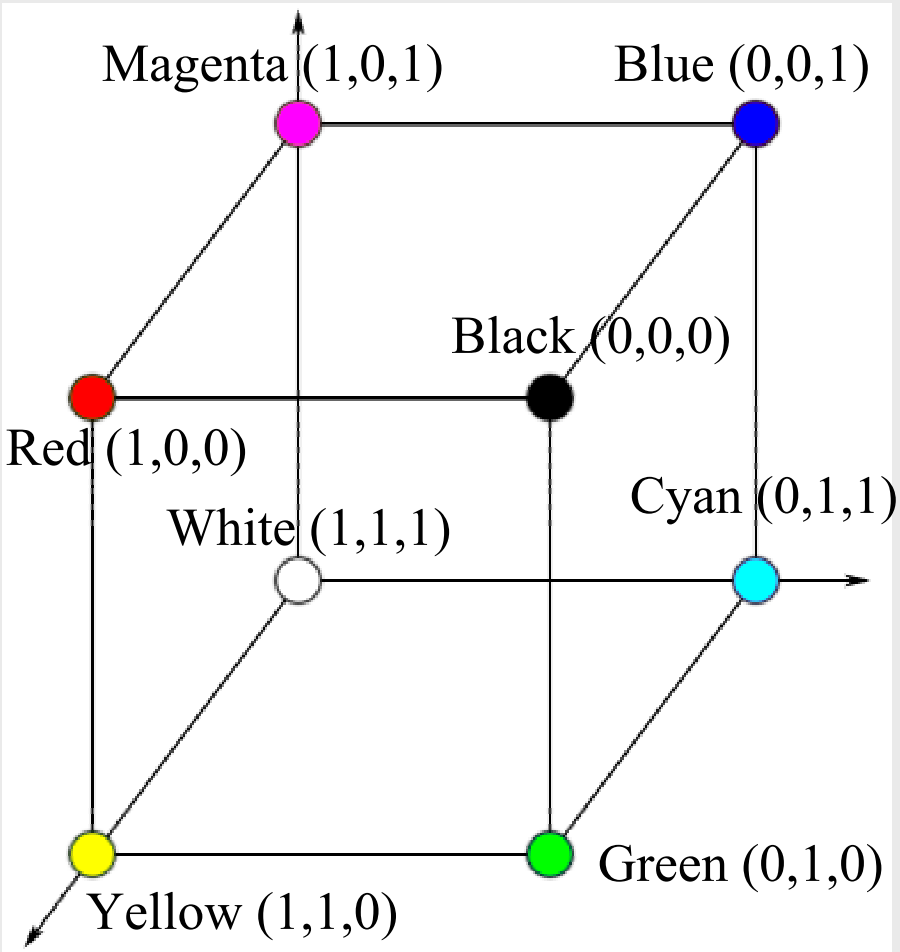
GRAPHICS & MEDIA LAB
VIDEO GROUP

CMYK (Cyan, Magenta, Yellow, blacK).





Расчет RGB, CMYK, CMY



RGB ➔ CMY

$C = 255 - R$

$M = 255 - G$

$Y = 255 - B.$

CMY➔CMYK

$K = \min(C,M,Y),$

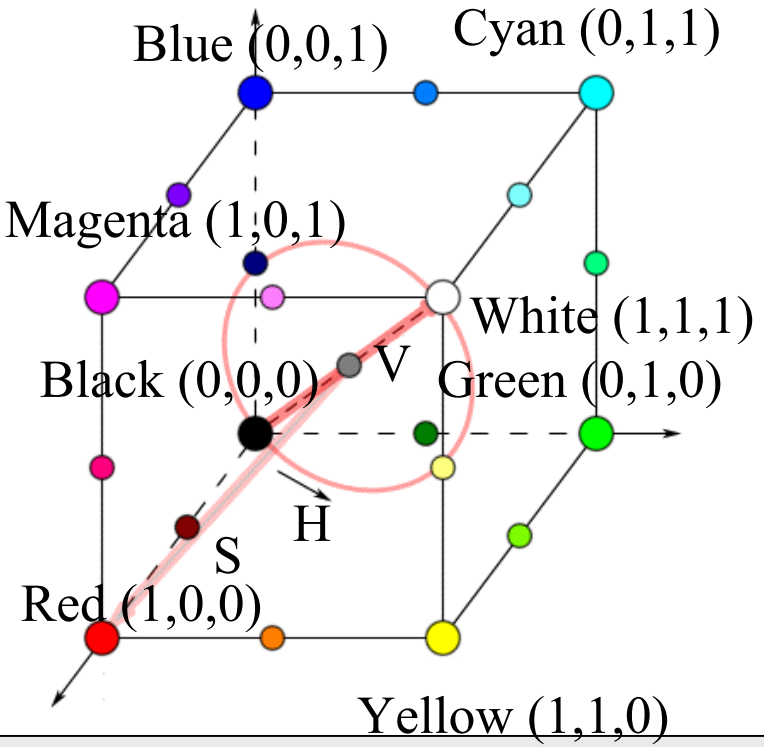
$C = C - K,$

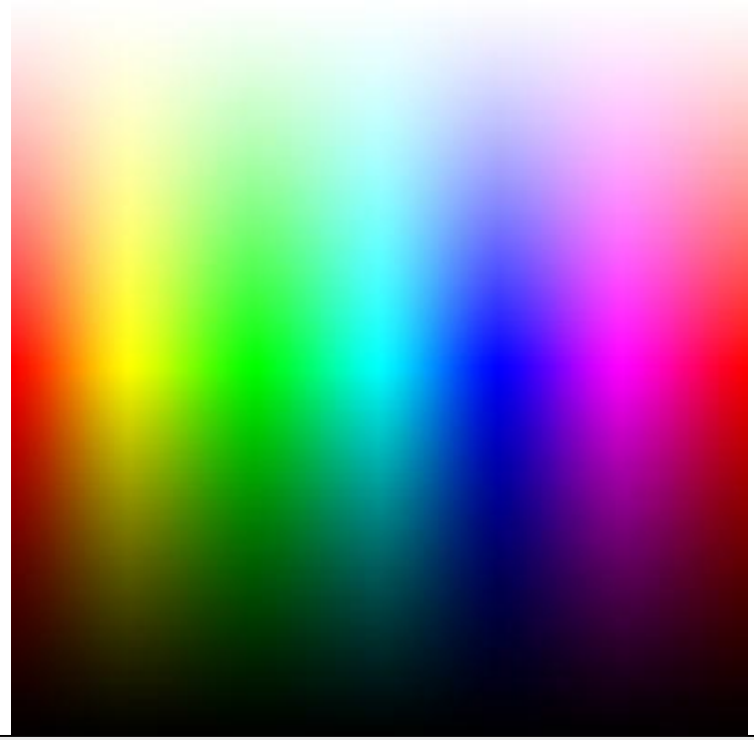
$M = M - K,$

$Y = Y - K.$

Пространство HSV

Модель **HSV (Hue, Saturation, Value).** Построена на основе субъективного восприятия цвета человеком.





Модель YUV

GRAPHICS & MEDIA LAB
VIDEO GROUP

$$Y = 0.299R + 0.587G + 0,114B$$

$$U = -0.147R - 0.289G + 0,436B$$

$$V = 0.615R + 0.515G + 0,100B = 0,877(R - Y)$$

$$R = Y + 1.140V$$

$$G = Y - 0.395U - 0.581V$$

$$B = Y + 2.032U$$

Модель YIQ

$$Y = 0.299*R + 0.587*G + 0.114*B$$

$$I = 0.596*R - 0.275*G - 0.321*B$$

$$Q = 0.212*R - 0.523*G + 0.311*B$$

$$R = Y + 0.956*I + 0.621*Q$$

$$G = Y - 0.272*I - 0.647*Q$$

$$B = Y - 1.107*I + 1.704*Q$$

Модель YCbCr (SDTV)

GRAPHICS & MEDIA LAB
VIDEO GROUP

$$Y = 0.299*R + 0.587*G + 0.114*B$$

$$Cb = -0.172*R - 0.339*G + 0.511*B + 128$$

$$Cr = 0.511*R - 0.428*G + 0.083*B + 128$$

$$R = Y + 1.371(Cr - 128)$$

$$G = Y - 0.698(Cr - 128) - 0.336(Cb - 128)$$

$$B = Y - 1.732(Cb - 128)$$

Классы изображений

GRAPHICS & MEDIA LAB
VIDEO GROUP

- **Класс 1. Изображения с небольшим количеством цветов** (4-16) и большими областями, заполненными одним цветом. Плавные переходы цветов отсутствуют. Примеры: деловая графика — гистограммы, диаграммы, графики и т.п.
- **Класс 2. Изображения, с плавными переходами цветов**, построенные на компьютере. Примеры: графика презентаций, эскизные модели в САПР, изображения, построенные по методу Гуро.
- **Класс 3. Фотореалистичные изображения.** Пример: отсканированные фотографии.
- **Класс 4. Фотореалистичные изображения с наложением деловой графики**. Пример: реклама.

Требования приложений к алгоритмам

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Высокая степень компрессии
- Высокое качество изображений
- Высокая скорость компрессии
- Высокая скорость декомпрессии
- Масштабирование изображений
- Возможность показать огрубленное изображение (низкого разрешения)
- Устойчивость к ошибкам
- Учет специфики изображения
- Редактируемость
- Небольшая стоимость аппаратной реализации. Эффективность программной реализации

Критерии сравнения алгоритмов

Невозможно составить универсальное сравнительное описание известных алгоритмов.

- **Худший, средний и лучший коэффициенты сжатия.**
- **Класс изображений**
- **Симметричность**
- **Есть ли потери качества?**
- **Характерные особенности алгоритма**

Алгоритм RLE

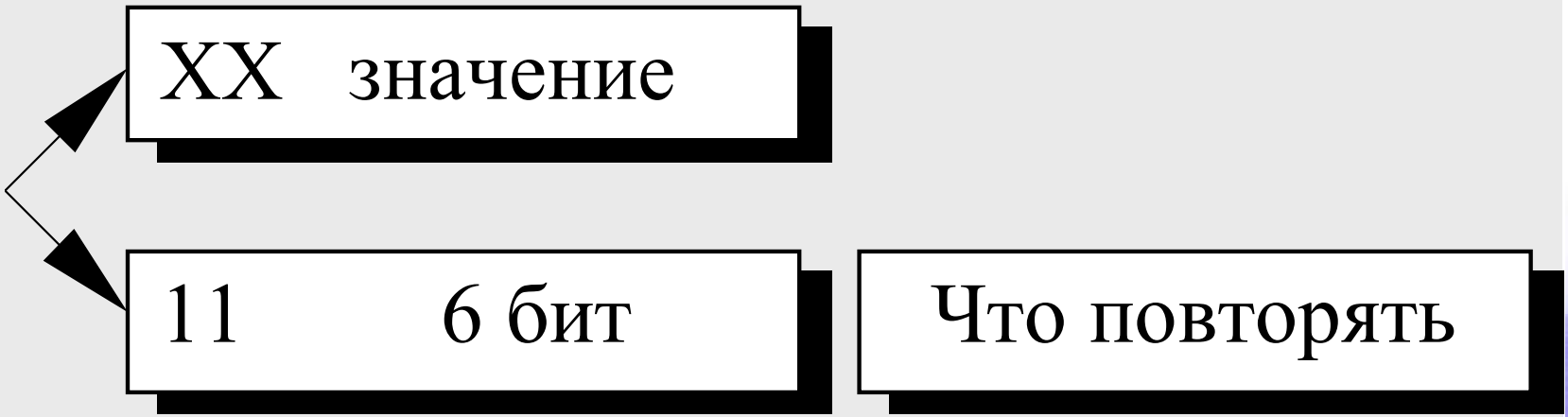
GRAPHICS & MEDIA LAB
VIDEO GROUP

Данный алгоритм необычайно прост в реализации. Групповое кодирование — от английского Run Length Encoding (RLE). Изображение в нем вытягивается в цепочку байт по строкам растра. Само сжатие в RLE происходит **за счет того, что в исходном изображении встречаются цепочки одинаковых байт**. Замена их на пары <счетчик повторений, значение> уменьшает избыточность данных.

RLE – Первый вариант

```
Initialization(...);
do {
   byte = ImageFile.ReadNextByte();
   if(является счетчиком(byte)) {
       counter = Low6bits(byte)+1;
       value = ImageFile.ReadNextByte();
       for(i=1 to counter)
             DecompressedFile.WriteByte(value)
   }
   else {
   DecompressedFile.WriteByte(byte)
} while(ImageFile.EOF());
```

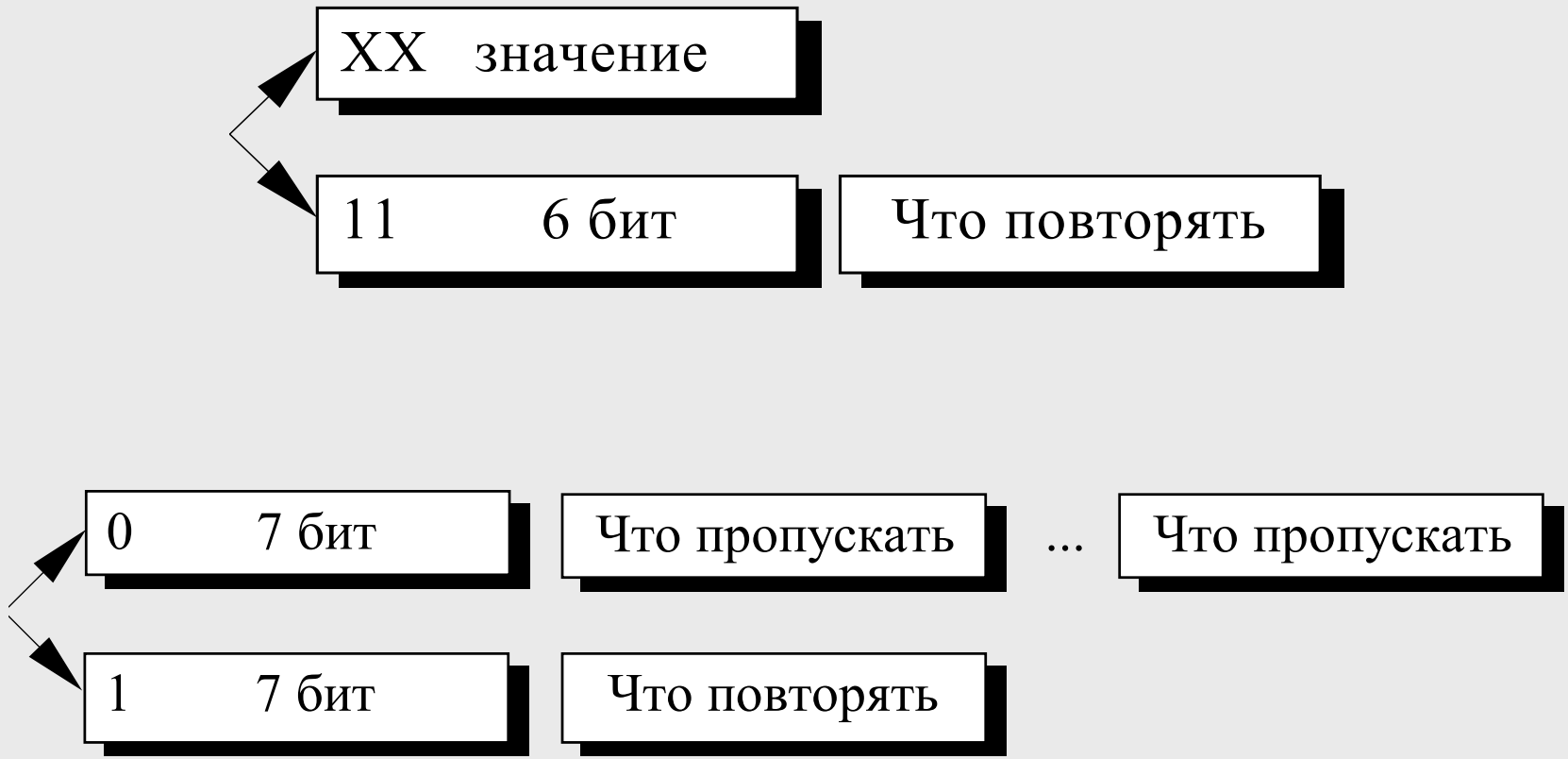
RLE – Первый вариант (схема)



RLE – Второй вариант

```
Initialization(...);
do {
   byte = ImageFile.ReadNextByte();
   counter = Low7bits(byte)+1;
   if(если признак повтора(byte)) {
        value = ImageFile.ReadNextByte();
        for (i=1 to counter)
               CompressedFile.WriteByte(value)
   }
   else {
        for(i=1 to counter){
               value = ImageFile.ReadNextByte();
               CompressedFile.WriteByte(value)
   }
   CompressedFile.WriteByte(byte)
} while(ImageFile.EOF());
```

RLE – Схемы вариантов



RLE – Характеристики

Коэффициенты компрессии: Первый вариант: 32, 2, 0,5. Второй вариант: 64, 3, 128/129. (Лучший, средний, худший коэффициенты)

Класс изображений: Ориентирован алгоритм на изображения с небольшим количеством цветов: деловую и научную графику.

Симметричность: Примерно единица.

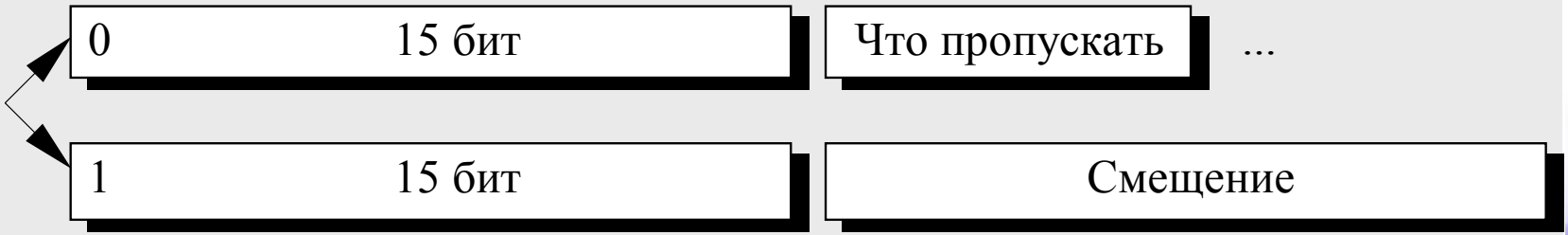
Характерные особенности: К положительным сторонам алгоритма, пожалуй, можно отнести только то, что он не требует дополнительной памяти при архивации и разархивации, а также быстро работает. Интересная особенность группового кодирования состоит в том, что степень архивации для некоторых изображений может быть существенно повышена всего лишь за счет изменения порядка цветов в палитре изображения.

Алгоритм LZW

GRAPHICS & MEDIA LAB
VIDEO GROUP

Название алгоритм получил по первым буквам фамилий его разработчиков — Lempel, Ziv и Welch. Сжатие в нем, в отличие от RLE, осуществляется уже за счет **одинаковых цепочек** байт.

Схема алгоритма LZ



LZW / Сжатие

```
InitTable();
CompressedFile.WriteCode(ClearCode);
CurStr=пустая строка;

while(не ImageFile.EOF()){              //Пока не конец файла
        C=ImageFile.ReadNextByte();
        if(CurStr+C есть в таблице)
        CurStr=CurStr+C;                //Приклеить символ к строке
   else {
        code=CodeForString(CurStr);     //code-не байт!
        CompressedFile.WriteCode(code);
        AddStringToTable (CurStr+C);
        CurStr=C;                       // Строка из одного символа
   }
}
code=CodeForString(CurStr);
CompressedFile.WriteCode(code);
CompressedFile.WriteCode(CodeEndOfInformation);
```

LZW / Пример

Пусть мы сжимаем последовательность 45, 55, 55, 151, 55, 55, 55.

“45” — есть в таблице;

“45, 55” — нет. Добавляем в таблицу <258>“45, 55”. В поток: <45>;

“55, 55” — нет. В таблицу: <259>“55, 55”. В поток: <55>;

“55, 151” — нет. В таблицу: <260>“55, 151”. В поток: <55>;

“151, 55” — нет. В таблицу: <261>“151, 55”. В поток: <151>;

“55, 55” — есть в таблице;

“55, 55, 55” — нет. В таблицу: “55, 55, 55” <262>. В поток: <259>;

Последовательность кодов для данного примера, попадающих в выходной поток: <256>, <45>, <55>, <55>, <151>, <259>.

LZW / Добавление строк

Код этой строки добавляется в таблицу

$C_n, C_{n+1}, C_{n+2}, C_{n+3}, C_{n+4}, C_{n+5}, C_{n+6}, C_{n+7}, C_{n+8}, C_{n+9},$

Коды этих строк идут в выходной поток

Таблица для LZW

0	‘0’
…	…
255	‘255’
256	ClearTable
257	EndOfInformation
258	
259	
…	
4095	

- Таблица состоит из 4096 строк.
- 256 и 257 являются служебными.
- 258 … 4095 содержат непосредственно сжимаемую информацию.

Пример – цепочка нулей

0	‘0’
…	…
255	‘255’
256	ClearTable
257	EndOfInformation
258	‘00’
259	‘000’
…	
4095	

Кол-во считываемых байт:

1 2 3

Общее число считанных байт:

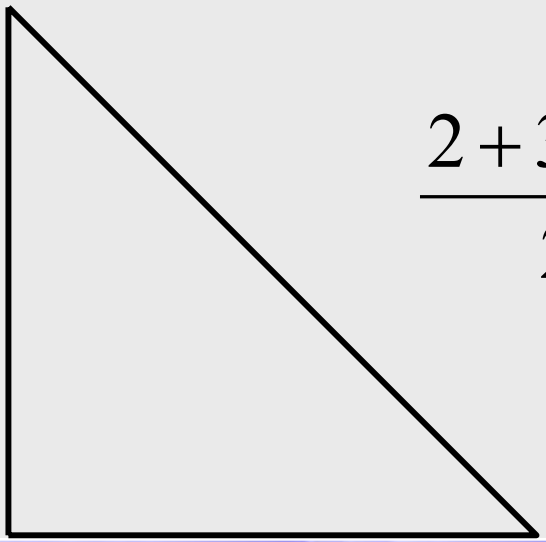
1 3 6

Информация заносится в стр.:

- 258 259

Степень сжатия цепочки нулей

0
..
255
256
257
258
259
..
4095



Рассчитываем арифметическую прогрессию:

$$Sn = \frac{a1 + an}{2} * n$$

$$\frac{2 + 3839}{2} * 3839 = 7372799$$

Наихудший случай

0	‘0’
…	…
255	‘255’
256	ClearTable
257	EndOfInformation
258	’21’
259	’13’
…	
4095	

Последовательность : 121314151617…

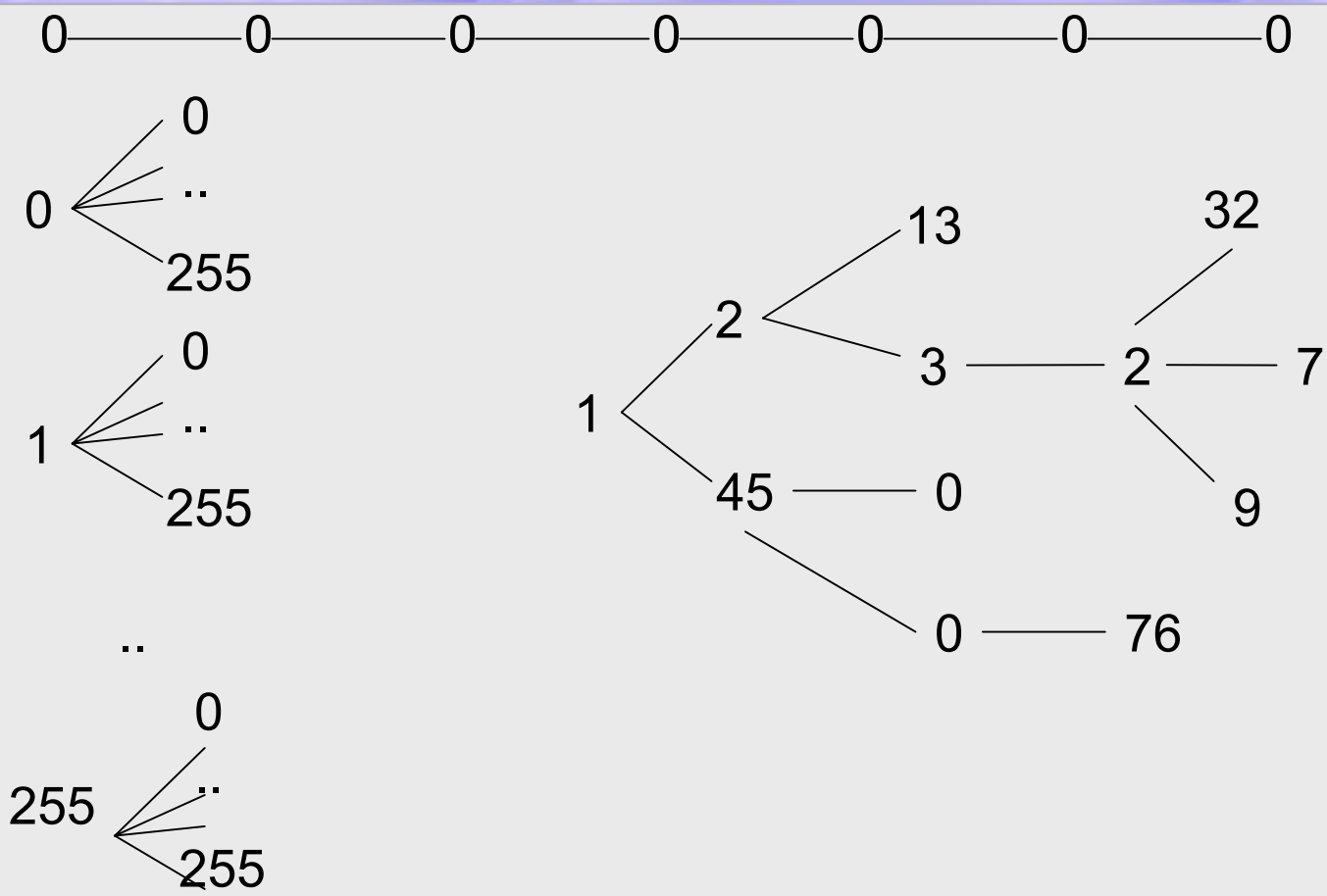
Мы видим, что у нас нет одинаковых цепочек даже из 2 символов => сжатия не происходит.

Степень сжатия наихудшего случая

0	‘0’
…	…
255	‘255’
256	ClearTable
257	EndOfInformation
258	’21’
259	’13’
…	
4095	

- Происходит увеличение файла в 1.5 раза. Т.к. мы ни разу не встретили подстроку, которая уже есть в таблице.

Таблица дерево



Пример

Последовательность: 45, 55, 55, 151, 55, 55, 55.
“45” – есть в таблице;
“45, 55” – нет. В таблицу: <258>”45, 55”. В поток:<45>
“55, 55” – нет. В таблицу: <259>”55, 55”. В поток:<55>
“55, 151” – нет. В таблицу: <260>”55, 151”. В поток:<55>
“151, 55” – нет. В таблицу: <261>”151, 55”. В поток:<151>
“55, 55” – Есть в таблице;
“55, 55, 55” – нет. В таблицу: <262>”55, 55, 55”. В поток:<295>

Итого в потоке: <256>,<45>,<55>,<55>,<151>,<259>.

Пример

0	‘0’
…	…
255	‘255’
256	ClearTable
257	EndOfInformation
258	’45, 55’
259	’55, 55’
260	’55, 151’
261	‘151, 55’
262	’55, 55, 55’

Последовательность:
45, 55, 55, 151, 55, 55, 55.

Итого в потоке:
<256>,<45>,<55>,<55>,
<151>,<259>.

LZW / Декомпрессия

```
code=File.ReadCode();
while(code != CodeEndOfInformation){
    if(code = ClearCode) {
          InitTable();
          code=File.ReadCode();
          if(code = CodeEndOfInformation)
                              {закончить работу};
          ImageFile.WriteString(StrFromTable(code));
          old_code=code;
    }
    else {
          if(InTable(code)) {
                    ImageFile.WriteString(FromTable(code));
                    AddStringToTable(StrFromTable(old_code)+
                              FirstChar(StrFromTable(code)));
                    old_code=code;
          }
          else {
                    OutString= StrFromTable(old_code)+
                       FirstChar(StrFromTable(old_code));
                    ImageFile.WriteString(OutString);
                    AddStringToTable(OutString);
                    old_code=code;
          }
    }
}
```

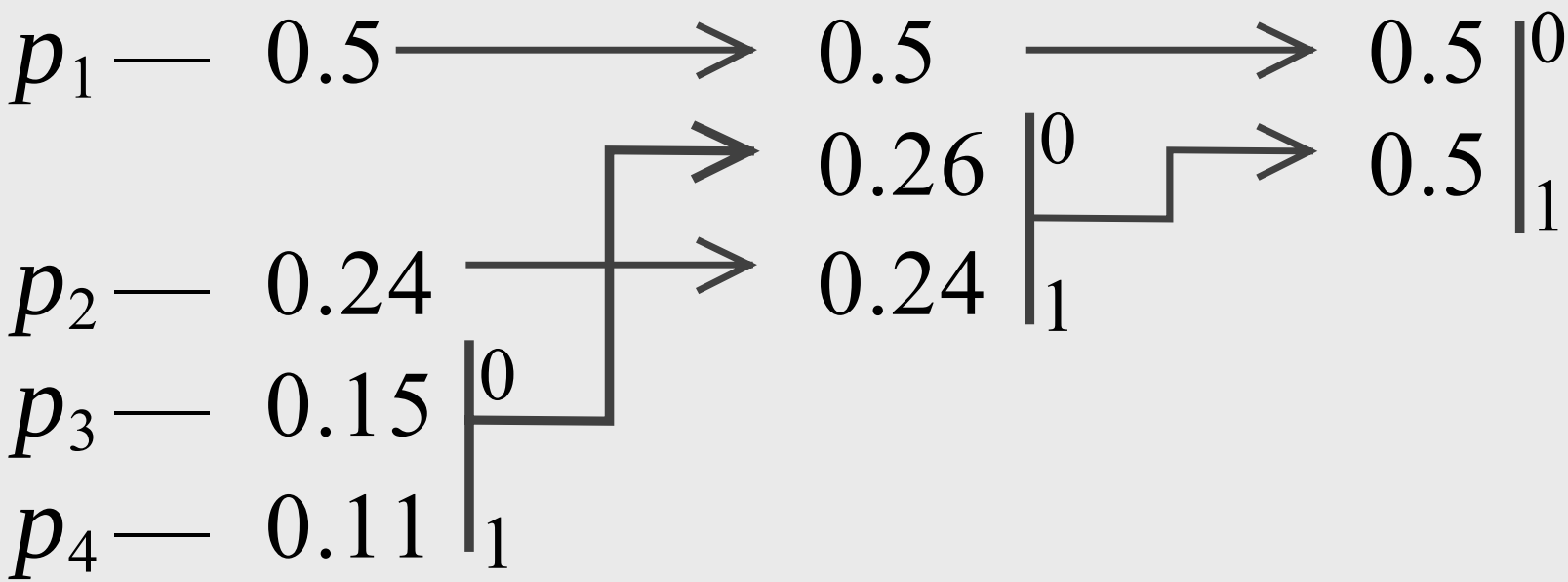
LZW / Характеристики

- **Коэффициенты компрессии:** Примерно 1000, 4, 5/7 (Лучший, средний, худший коэффициенты). Сжатие в 1000 раз достигается только на одноцветных изображениях размером кратным примерно 7 Мб.
- **Класс изображений:** Ориентирован LZW на 8-битные изображения, построенные на компьютере. Сжимает за счет одинаковых подцепочек в потоке.
- **Симметричность:** Почти симметричен, при условии оптимальной реализации операции поиска строки в таблице.

Алгоритм Хаффмана

Использует только частоту появления одинаковых байт в изображении. Сопоставляет символам входного потока, которые встречаются большее число раз, цепочку бит меньшей длины. И, напротив, встречающимся редко — цепочку большей длины. Для сбора статистики требует двух проходов по изображению.

Алгоритм Хаффмана-2



Алгоритм Хаффмана-3

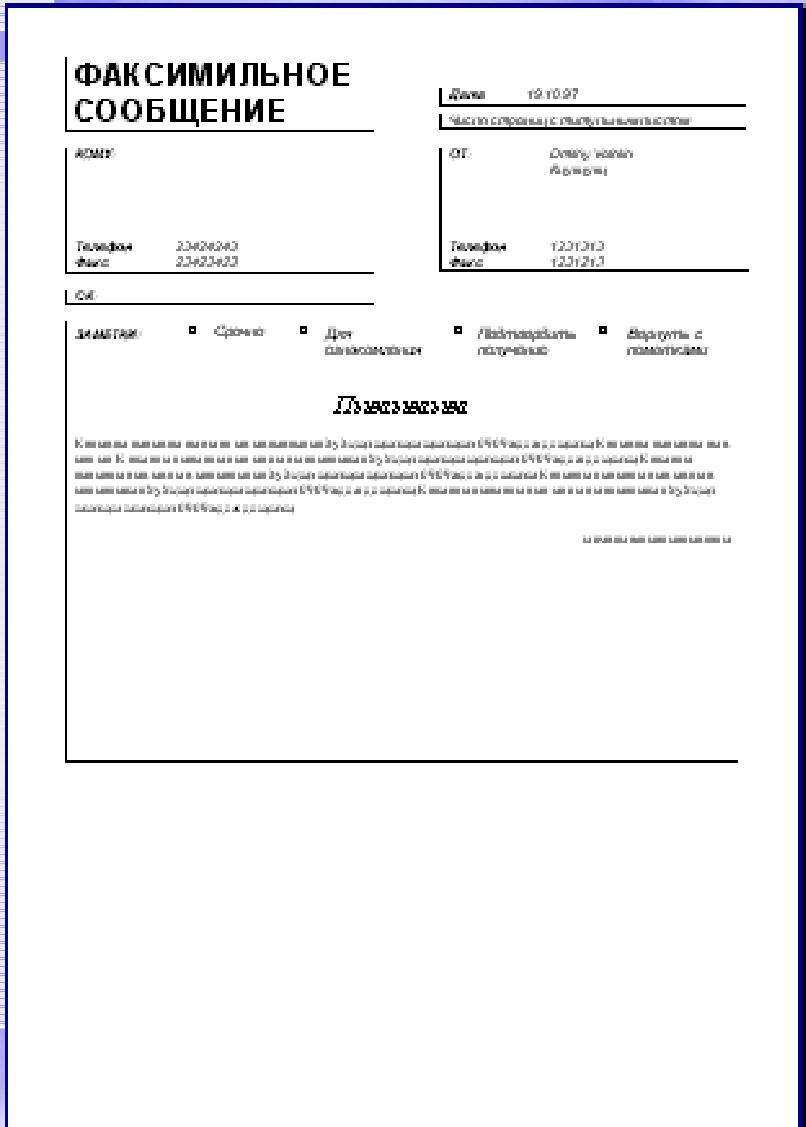
GRAPHICS & MEDIA LAB
VIDEO GROUP

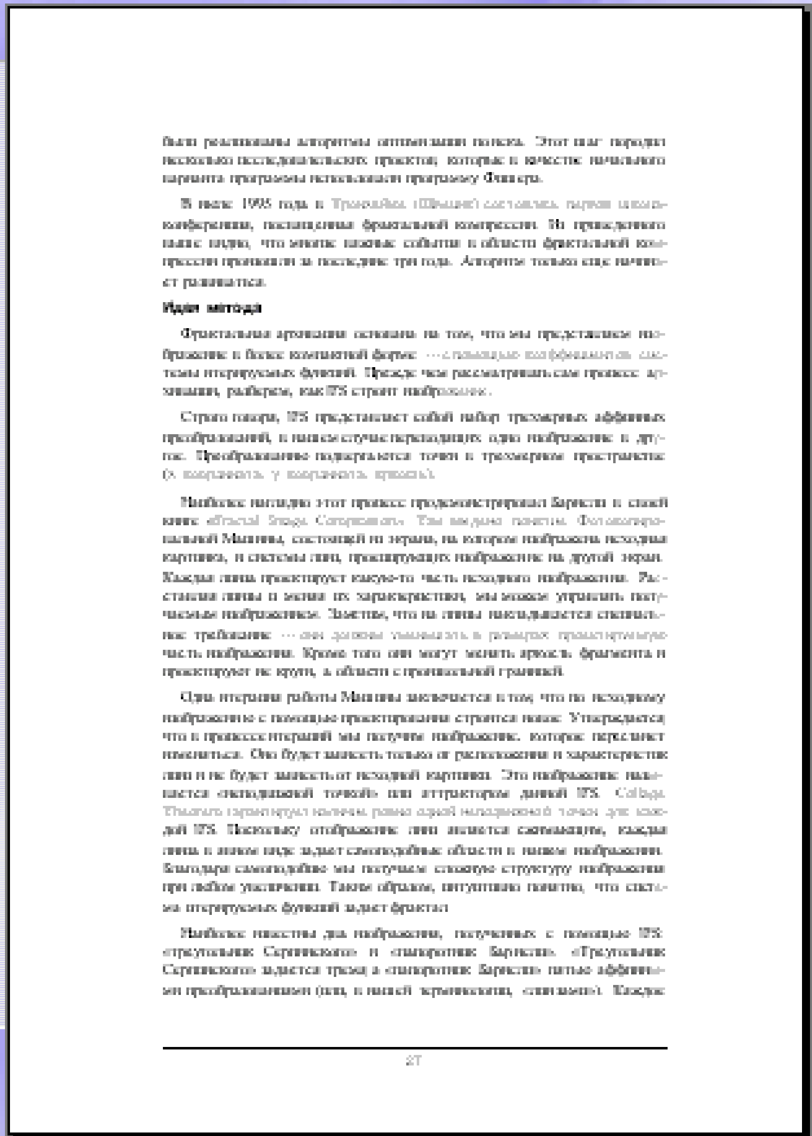
- **Коэффициенты компрессии:** 8, 1,5, 1 (Лучший, средний, худший коэффициенты).
- **Класс изображений:** Практически не применяется к изображениям в чистом виде. Обычно используется как один из этапов компрессии в более сложных схемах.
- **Симметричность:** 2 (за счет того, что требует двух проходов по массиву сжимаемых данных).
- **Характерные особенности:** Единственный алгоритм, который не увеличивает размера исходных данных в худшем случае (если не считать необходимости хранить таблицу перекодировки вместе с файлом).

CCITT Group 3

Последовательности подряд идущих черных и белых точек в нем заменяются числом, равным их количеству. А этот ряд, уже в свою очередь, сжимается по Хаффману с фиксированной таблицей.

Примеры факсов





Алгоритм CCITT G3

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Последовательности подряд идущих черных и белых точек заменяются числом, равным их количеству.
- Этот ряд сжимается по Хаффману с фиксированной таблицей.
- Каждая строка сжимается независимо, если строка начинается с черной точки, то считаем, что она начинается белой серией длиной 0.Например, последовательность длин серий 0, 3, 556,10,.. означает, что в строке идут сначала 3 черных, 556 белых, 10 черных точек и т.д.

Алгоритм компрессии:

```
For (по всем строкам изображения) {
    Преобразуем строку в набор длин серий;
    for (по всем сериям) {
        if (серия белая) {
             L = длина серии;
             while (L > 2623) { // 2623 = 2560 + 63
                    L -= 2560; Записать белый код для (2560);
                  }
             if (L > 63) {
                    L2 = МаксимальныйСостКодМеньшеL(L);
                    L -= L2; Записать белый код для (L2)
             };
             ЗаписатьБелыйКодДля(L); // код завершения
        } else  {
          // аналогично для черных серий
         ...}
```

Пример работы алгоритма

GRAPHICS & MEDIA LAB
VIDEO GROUP

В терминах регулярных выражений для каждой строки изображения выходной битовый поток вида:

((<Б-2560>)*[<Б-сст.>]<Б-зв>(<Ч-2560>)*[<Ч-сст>]<Ч-зв>)+[(<Б-2560>)*[<Б-сст.>]<Б-зв.>],где:

()* - повтор 0 или более раз, ()+ - повтор 1 или более раз, [] – включение 1 или 0 раз.

Для примера 0, 3, 556, 10,… ,будет сформирован

код: <Б-0><Ч-3><Б-512><Б-44><Ч-10> или

Согласно таблице:

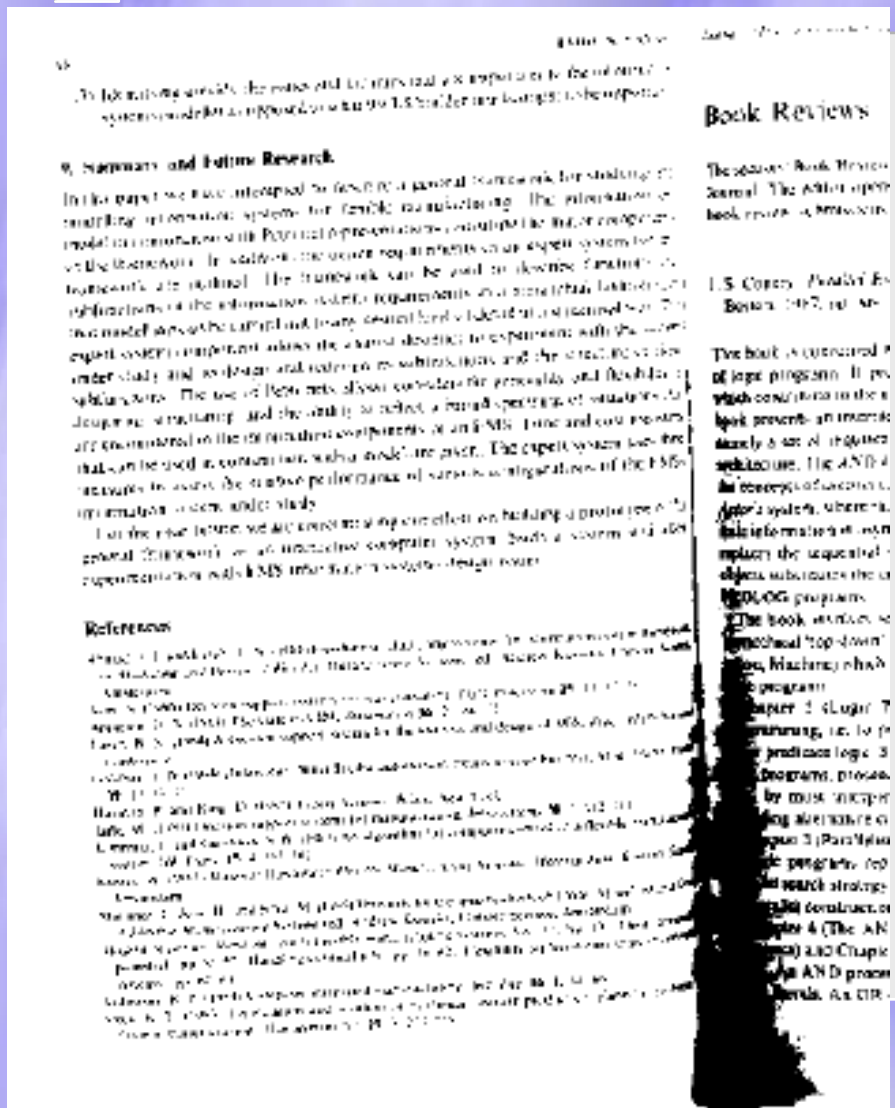
00110101 10011001 01001011 010000100

Для приведенной строки в 569 бит полусен код длиной в 33 бита, т.е. Коэфф сжатия – 17 раз

Таблица кодов завершения

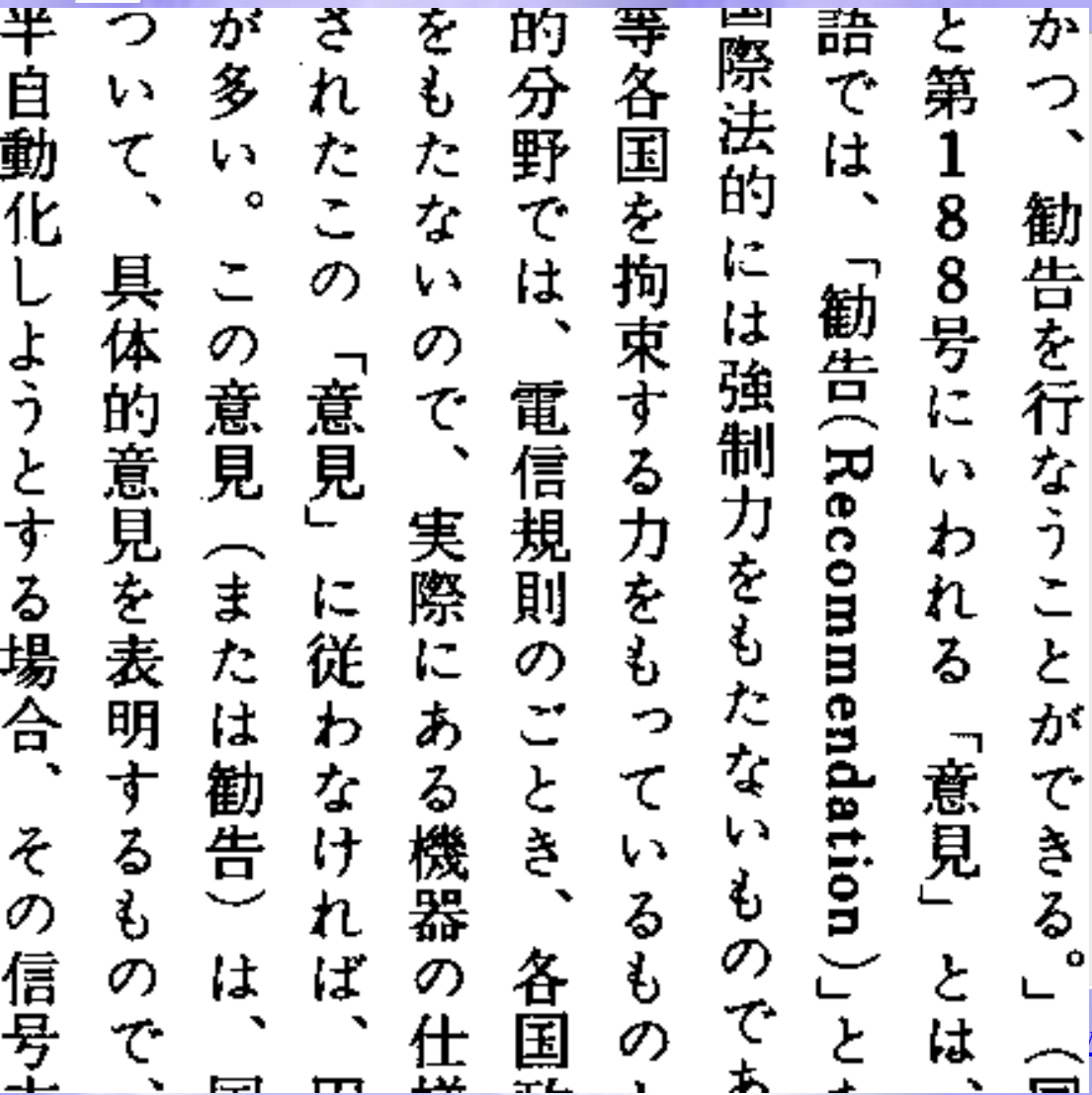
Длина серии	Код белой подстроки	Код черной подстроки строки
0	00110101	0000110111
1	00111	010
2	0111	11
3	1000	10
4	1011	011
5	1100	0011
6	1110	0010
7	1111	00011

Проблемы при сжатии



Пример, когда часть страницы идет под косым углом + разворот книги темный

Проблемы при сжатии


かつ、勧告を行なうことができる。」（同
と第188号にいわれる「意見」とは、
語では、「勧告（Recommendation）」と
国際法的には強制力をもたないものであ
等各国を拘束する力をもっているもの
的分野では、電信規則のごとき、各国
をもたないので、実際にある機器の仕様
されたこの「意見」に従わなければ、
が多い。この意見（または勧告）は、
ついて、具体的意見を表明するもので、
半自動化しようとする場合、その信号

Пример факса (часть текста рекомендаций стандарта CCITT) на японском (?) языке.

CCITT Group 3 / Характеристики

GRAPHICS & MEDIA LAB
VIDEO GROUP

- **Коэффициенты компрессии:** лучший коэффициент стремится в пределе к 213.(3), средний 2, в худшем случае увеличивает файл в 5 раз.
- **Класс изображений:** Двуцветные черно-белые изображения, в которых преобладают большие пространства, заполненные белым цветом.
- **Симметричность:** Близка к 1.
- **Характерные особенности:** Данный алгоритм чрезвычайно прост в реализации, быстр и может быть легко реализован аппаратно.