Лабораторная работа: **создание видеофайлов в среде MATLAB.**

Цель работы – изучение функций, использующихся для создания изображений и видеоизображений в среде Matlab.

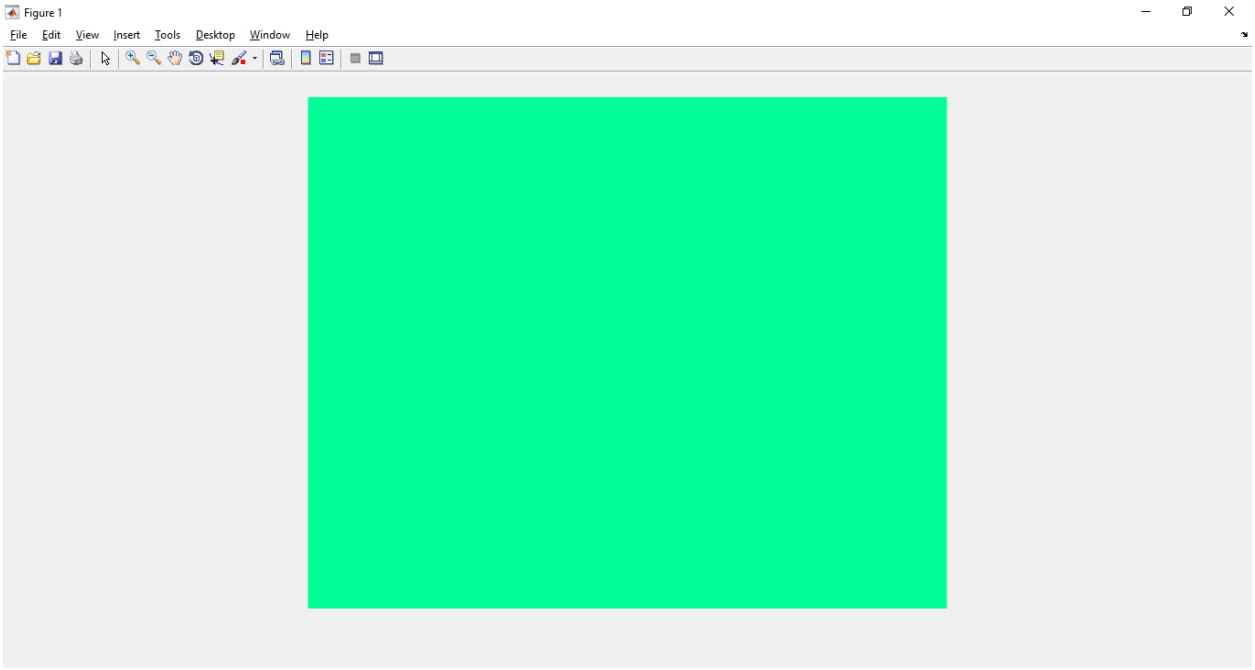
1. Краткие теоретические сведения.
   Телевизионное изображение стандартного качества представляет собой видеопоток с разрешением 720x576 пикселей и частотой кадров 25. Зная эти параметры, можно нарисовать в среде Matlab любую картинку в формате RGB, задавая цвет каждому пикселю. Самый простой пример создания изображения является заполнение экрана (всех пикселей) одним цветом. Для этого необходимо инициализировать в Matlab 2 цикла: по строке и по столбцу. Заполняя каждый пиксель равными значениями в диапазоне от 0 (черный) до 1 (белый).
   Пример данной реализации будет (файл field.m)

```
clear;clc;                      %Очистка памяти в Матлаб
weight=720;                     %Задание высоты изображения
height=576;                     %Задание ширины изображения
color=0.5;                      %Задание серого цвета 0 - черный, 1 - белый
b=zeros(height,weight,3);       %Создания массива нулей в формате RGB
                                %изначально всё поле черное)
for i=1:1:height                %Инициализация цикла по ширине (по строкам)
    for j=1:1:weight            %Инициализация цикла по высоте (по столбцам)
        for c=1:1:3             %Инициализация цикла по RGB компонентам
     b(i,j,c)=color;            %Задаём для каждой цветовой компоненты один и
                                %тот же цвет, получая определенный оттенок
                                %серого.
        end;
    end;
end;
figure(1);imshow(b);            %выводим на экран полученное изображения
```

Стоит заметить, что тут можно выбирать любой размер изображения и любой оттенок серого. Также при желании можно заменить 3-ий цикл на «ручное» назначения каждой цветовой компоненте определенного цвета:

```
b(i,j,1)=0;
b(i,j,2)=255;
b(i,j,3)=150;
```

В результате данного примера мы получим светло-зеленое поле.

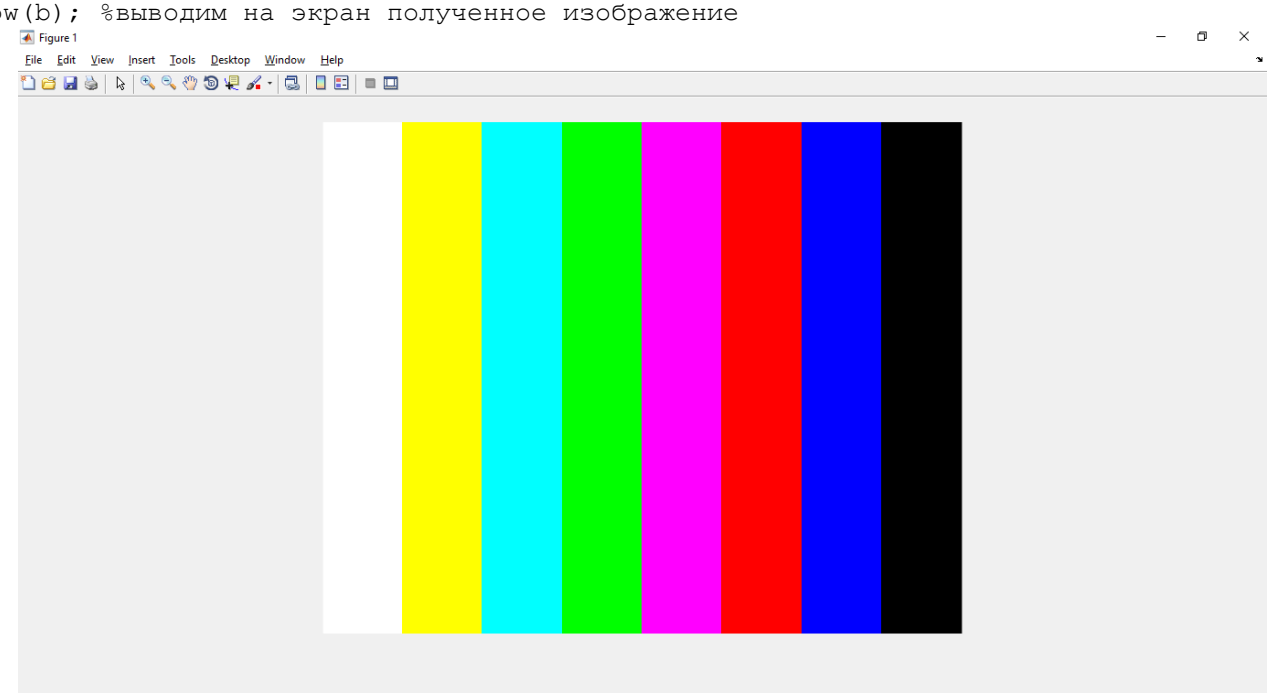


2) создание изображение цветных полос (файл bars-avi.m до строки 54)

```
clear;clc;                          %очистка памяти
height=576;                         %задаём ширину и высоту изображения
weight=720;
bar_weight=weight/8;                %делим ширину изображения на 8 равных
                                    отрезков, в данном случае по 90 пикселов
x=uint8(255);                       %задаём максимальную насыщенность
                                    %компонент
b=uint8(zeros(height,weight,3));    %заполняем изображение чёрным полем, путём
%создания матрицы нулей, которую далее будем заполнять.

%запускаем цикл по элементам строк и столбцов, каждый элемент соответствует
%одному пикселу.
for i=1:1:height
     for j=1:1:weight
          if (j>0 && j<=bar_weight*1)
%первый столбец белого цвета, в палитре RGB все цветовые компоненты равны 255
               b(i,j,1)=x;
               b(i,j,2)=x;
               b(i,j,3)=x;
           end;
           if (j>=bar_weight*1 && j<=bar_weight*2)
               b(i,j,1)=x;
               b(i,j,2)=x;
               b(i,j,3)=0;
           end;
           if (j>=bar_weight*2 && j<=bar_weight*3)
               b(i,j,1)=0;
               b(i,j,2)=x;
               b(i,j,3)=x;
           end;
           if (j>=bar_weight*3 && j<=bar_weight*4)
               b(i,j,1)=0;
               b(i,j,2)=x;
               b(i,j,3)=0;
```

```
            end;
            if (j>=bar_weight*4 && j<=bar_weight*5)
                b(i,j,1)=x;
                b(i,j,2)=0;
                b(i,j,3)=x;
            end;
            if (j>=bar_weight*5 && j<=bar_weight*6)
                b(i,j,1)=x;
                b(i,j,2)=0;
                b(i,j,3)=0;
            end;
            if (j>=bar_weight*6 && j<=bar_weight*7)
                b(i,j,1)=0;
                b(i,j,2)=0;
                b(i,j,3)=x;
            end;
            if (j>=bar_weight*7 && j<=bar_weight*8)
                b(i,j,1)=0;
                b(i,j,2)=0;
                b(i,j,3)=0;
            end;
    end;
end;
imshow(b); %выводим на экран полученное изображение
```



Для сохранения полученного изображения на компьютере, достаточно в Command Window написать **`imwrite(b,'C:\temp\bars.jpg');`** где C:\temp – путь сохраняемого файла.

Цикл работает следующим образом:
Переменная i указывает на номер строки от 1 до 576. После того, как была выбрана строка начинается перебор всех элементов на строке от 1 до 720 – переменная j. Дальше с помощью оператора if производится сравнение текущего положения счётчика цикла. Цветные полосы следуют друг за другом через каждые 90 пикселей, поэтому необходимо задать условия, что делать в каждом конктретном случае.

`if (j>0 && j<=bar_weight*1)` означает, что если J больше нуля и меньше 90 (bar_weight=weight/8, что при данной ширине 720 пикселей равняется 90). Соответственно

bar_weight*2 = 180 и т.д. Дальше только остаётся задать для каждой цветовой компоненты её значение.

3) создание видеофайла (файл bars-avi.m, строки 56-67)

Чтобы перевести картинку в видео, необходимо добавить следующий код к текущему скрипту:

```
[X,map]=rgb2ind(b,65536);     %переводим изображение в индексное
F(1)=im2frame(X,map);   %создаём 1 кадр видео из полученной ранее картинки
for i=2:1:100           %копируем из 1 кадра во все остальные кадры (100 для
                        %примера, при 25 кадрах = 4 секунды видео)
    F(i)=F(1);
end;

v=VideoWriter('bars.avi','Uncompressed AVI'); %создаём сам видеофайл в
                                              %формате AVI
v.FrameRate=25;               %задаём частоту кадров (по умолчанию 30)
open(v);                      %открывает созданный контейнер для работы
writeVideo(v,F);              %записываем в контейнер наши кадры
close(v);                     %закрываем видео.
```

Таким образом мы создали компьютерным путём тестовое видео, в котором каждый пиксель описан с помощью программного кода.

В заключении можно выполнить команду `info=get(v)` и посмотреть основные параметры видеоизображения.

```
info =
             Duration: 4
             Filename: 'bars.avi'
                 Path: 'C:\Matlab\bars'
           FileFormat: 'avi'
               Height: 576
VideoCompressionMethod: 'None'
                Width: 720
        ColorChannels: 3
           FrameCount: 100
          VideoFormat: 'RGB24'
     VideoBitsPerPixel: 24
            FrameRate: 25
```

4) просмотр графика цветовых компонент (подобие осциллограммы) (файл plot_waveform.m)

Самым простым способом посмотреть осциллограмму будет построить график любой одной строки, так как все строки содержат абсолютно одинаковую информацию. Далее можно будет построить график по полученному вектору. Если строить на одной фигуре (на одном графике), то общая картина будет непонятна. Поэтому предлагается следующий код для вывода графика на экран. Лучше всего будет его создать отдельным m-файлом. Особую важность стоит уделить преобразованию компоненты яркости. Для этого необходимо будет сделать преобразование из цветного изображения в чёрно-белое (оттенки серого). Сделать это можно двумя способами: вручную с помощью формулы: Y = 0.299R + 0.587G + 0.144B, либо с помощью команды BlackWhite = rgb2gray (x) , где x – имя массива RGB.

```
bw1=uint8(zeros(height,weight));     %инициализируем пустой двумерный массив
for h=1:1:height                     %запускаем цикл по строкам
w=1:1:weight;                        %создаем вектор элементов по длине строки
%далее в массив bw(1) будут записываться суммарное значение яркости
%исходя из формулы перевода
bw1(h,w)=uint8(((b(h,w,1)*0.299))+((b(h,w,2)*0.587))+((b(h,w,3)*0.114)));
```

```
end;
bw=rgb2gray(b);          %переводим с помощью встроенной функции
j=100;                   %где 100 - 100-я строка изображения (может быть любое
число)
r(w)=b(j,w,1);           %создаём вектор красной компоненты
figure(2);plot(r, '-r');%выводим график красного цвета для красной компоненты
title('красная компонента');
g(w)=b(j,w,2);           %создаём вектор зелёной компоненты
figure(3);plot(g, '-g');%выводим график зелёного компоненты
title('зеленая компонента');
bl(w)=b(j,w,3);          %создаём вектор синей компоненты
figure(4);plot(bl,'-b');%выводим график синей компоненты
title('синяя компонента');
bw1i(w)=bw1(j,w);        %создаём вектор "яркостного "сигнала", полученного
ручным способом
figure(5); plot(bw1i);  %выводим график
title('яркостная компонента, рассчитанная по формуле');
bwi(w)=bw(j,w);          %создаём вектор "яркостного "сигнала", полученного
встроенной функцией
figure(6);plot(bwi);     %выводим график
title('яркостная компонента через MATLAB');
figure(7);imshow(bw);    %выводим чёрно-белое изображение, полученное с
помощью встроенной функции
figure(8);imshow(bw1);   %выводим чёрно-белое изображение, полученное ручным
способом
```

`VideoWriter` objects позволяет создавать видео файлы из массимово или MATLAB® movies.

Можно использовать `VideoWriter` функцию с предустановленным профилем или создать `VideoWriter` с параметрами связанными для определенного формата.

`v = VideoWriter(filename,profile)` создаёт объект videowriter и применяет установленные свойства, связанные с определённым форматом файла (такие как MPEG-4 или `'Uncompressed AVI'`).

`VideoWriter` позволяет создать видеофайлы можно со следующими расширениями

`.avi`	AVI
`.mj2`	Motion JPEG 2000
`.mp4` или `.m4v`	MPEG-4 (Windows® 7 и новее, или Mac OS X 10.7 и новее)

Тип файла описывает одной из следующих строк

Value of `profile`	**Description**
`'Archival'`	Motion JPEG 2000 файл с компрессией без потерь
`'Motion JPEG AVI'`	AVI файл с кодированием Motion JPEG
`'Motion JPEG 2000'`	Motion JPEG 2000
`'MPEG-4'`	MPEG-4 файл с кодированием H.264 (Windows® 7 и новее, или Mac OS X 10.7 и новее)
`'Uncompressed AVI'`	Несжатый AVI файл с `RGB24`video
`'Indexed AVI'`	Несжатый AVI файл с индексным видео
`'Grayscale AVI'`	Несжатый AVI файл с видео с оттенками серого

`profile` устанавливают значения по умолчанию для видео параметров такие как метод сжатия.

Формат файла

FileFormat - тип файла для записи `'avi'|'mp4'|'mj2'`.

Filename – имя файл

FrameCount – число кадров, которые записаны в видеофайл (целое число)

FrameRate - частота кадров в секунду (по умолчания 30). После инициализации невозможно изменить. Можно установить любое положительное число.

Типы данных: `single|double|int8|int16|int32|int64|uint8|uint16|uint32|uint64`

Height – высота каждого кадра в пикселях, указанная как скалярная величина. `writeVideo` устанавливает высоту и ширину на основе первого кадра.

MPEG-4 файлы требуют четного значения. Если значением будет нечётное число, VideoWriter добавит снизу чёрную строку. Для MPEG-4 в системах Windows значения высоты могут находить от 64 до 1088 пикселей.

LosslessCompression - компрессия без потерь (`true` | `false`)

компрессия без потерь применяется только при записи Motion JPEG 2000 файлов.

Если `LosslessCompression` установлено как 'true', тогда:

- Функция writeVideo запишет данные, которые в записанном файле не будут отличаться от исходных данных
- VideoWriter проигнорирует любые установленные значения CompressionRatio

По умолчанию `LosslessCompression` стоит значение `false` для профиля `'Motion JPEG 2000'` и `true` для профиля `'Archival'` .

MJ2BitDepth - битовая глубина для файлов Motion JPEG 2000 (целое число от 1 до 16).

Если не задаётся отдельно, то по умолчанию берётся значение из исхоных данных, т.е. если формат данных в массиве был uint8 или int8 , то тогда MJ2BitDepth = 8.

Типы данных: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` |`uint32` | `uint64`

Path - полный путь до видеофайла.

Quality – видео качество (от 0 до 100, по умолчанию 75).

Чем больше значение, тем лучше качество видео и больше файлы. Quality допустимо использовать только для объектов `MPEG-4` или `Motion JPEG AVI` профиля.

VideoBitsPerPixel – количество бит на один пиксель.

AVI файлы с truecolor, Motion JPEG AVI и MPEG-4 используют 24 бита на пиксель (8 бит для каждой цветовой составляющей). Индексные и изображения с оттенками серого используют 8 бит на пиксель.

Для Motion JPEG 2000 файлов, число бит на пиксель зависит от значения `MJ2BitDepth` и количества цветовых компонент в данных изображения. Например, если исходные данные у `writeVideo` трехмерных массив с `uint16` или `int16`, значение по умолчанию `MJ2BitDepth` =16, значит `VideoBitsPerPixel` = 48.

Тип данных: `double`

VideoCompressionMethod - типы компрессии видео может принимать одно из следующих значений: `'None'` | `'H.264'` | `'Motion JPEG'` | `'Motion JPEG 2000'`.

VideoFormat - формат представления MATLAB

Для типов файлов, отличных от Motion JPEG 2000, `VideoFormat` принимает одно из следующих значений

Video Format	**Value of `VideoFormat`**
Uncompressed AVI, Motion JPEG AVI, or MPEG-4 files	`'RGB24'`
AVI files with indexed video	`'Indexed'`
AVI files with grayscale video	`'Grayscale'`

Для Motion JPEG 2000 файлов, `VideoFormat` зависит от значения `MJ2BitDepth` и от форматаа исходных данных `writeVideo`. Например, если вы не указали `MJ2BitDepth` параетр, `VideoWriter` устанавливает формат как в таблице.

Format of Image Data	Value of `VideoFormat`
Single-band `uint8`	`'Mono8'`
Single-band `int8`	`'Mono8 Signed'`
Single-band `uint16`	`'Mono16'`
Single-band `int16`	`'Mono16 Signed'`
Three-banded `uint8`	`'RGB24'`
Three-banded `int8`	`'RGB24 Signed'`
Three-banded `uint16`	`'RGB48'`
Three-banded `int16`	`'RGB48 Signed'`

Width – ширина каждого кадра в пикселях, указанная как скалярная величина. `writeVideo` устанавливает высоту и ширину на основе первого кадра.

MPEG-4 файлы требуют четного значения. Если значением будет нечётное число, VideoWriter добавит справа чёрный столбец. Для MPEG-4 в системах Windows значения высоты могут находить от 64 до 1920 пикселей.

Тип данных: `double`

Функции объекта

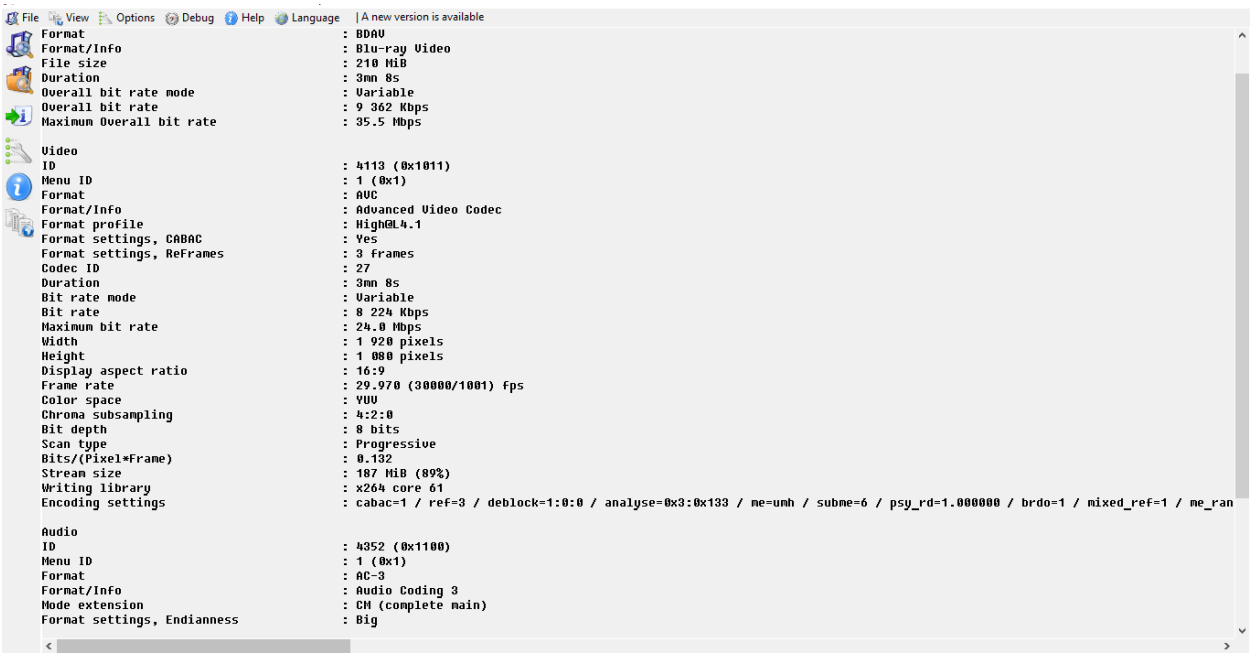
`open`	Открывает файл для записи видеоданных
`close`	Закрывает файл после записи видеоданных
`writeVideo`	Записывает видеоданных в файл
`VideoWriter.getProfiles`	Профили и форматы файлов, которые поддерживает VideoWriter

Степень сжатия.

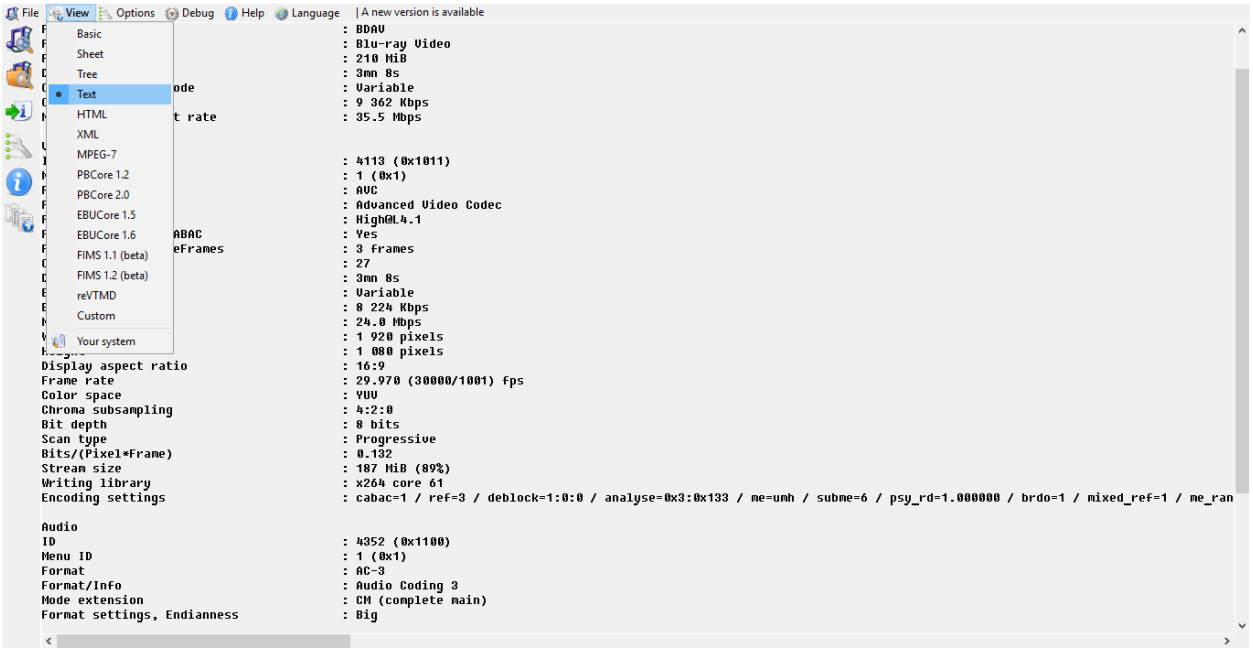
Степень сжатия устанавливается целым числом >1. Степень скажетия это отношенеи между количеством байт исходного изображения и сжатого. Видео данные сжимаются насколько возможно.

`CompressionRatio` доступна только для объектов, которые записываются в Motion JPEG 2000 файлы. После инициализации записи видеофайла невозможно изменить степень сжатия.

Для дополнительного контроля параметров видео удобно использовать программу MediaInfo. Она может показать все имеющиеся метаданные в видеофайле: формат, размер, скорость потока, соотношение сторон, количество кадров, применённый кодек и т.д..



Установить её достаточно просто. После установки и её первого запуска необходимо выбрать в настройках вид отображения и язык. Удобнее поставить вид отображения **TEXT ,** а язык выбрать английский, чтобы не возникало проблем с понимаем определённых терминов.



Вызвать данную страницу весьма просто: достаточно щёлкнуть правой кнопкной мыши на интересующем файле и выбрать **MediaInfo**.