

# **ImageMagick & ffmpeg**

## **Компьютерная графика**

Юдинцев В. В.

Кафедра теоретической механики

22 ноября 2024 г.



**САМАРСКИЙ УНИВЕРСИТЕТ**  
SAMARA UNIVERSITY

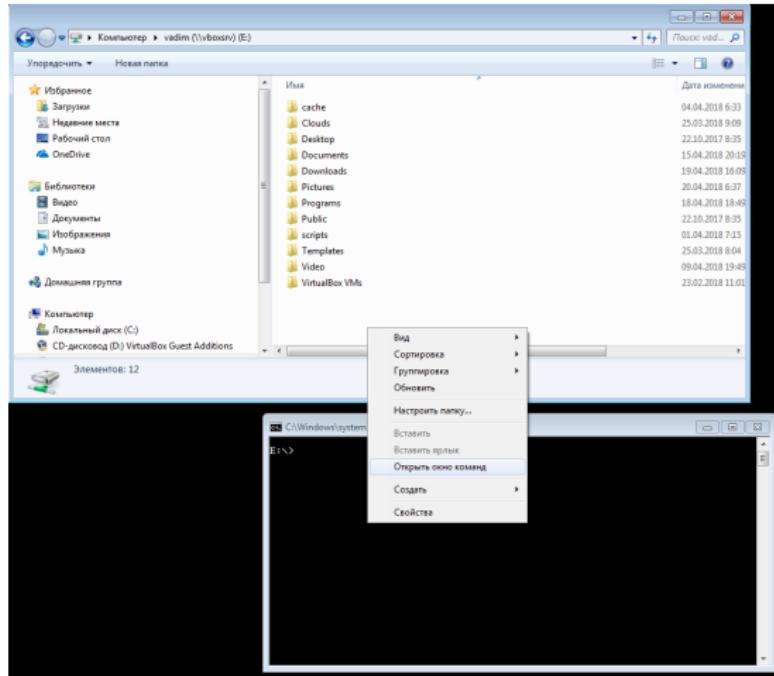
# Содержание

- 1 Работа с командной строкой
- 2 ImageMagick
- 3 FFMPEG

# Работа с командной строкой

# Командная строка

Открыть командную строку в текущем каталоге проводника: **Shift+ПКМ**



# ImageMagick

- **ImageMagick** – набор программ командной строки для преобразования, создания графических файлов.
- **ImageMagick** поддерживает около 200 форматов файлов.

## Основные утилиты

- `convert` – преобразование
- `mogrify` – преобразование с заменой исходного файла (файлов)
- `montage` – монтаж
- `compose` – совмещение (композиция) изображений

# Установка

<https://www.imagemagick.org/script/download.php#windows>

## Windows Binary Release

ImageMagick runs on Windows 10 (x86 & x64), Windows 8 (x86 & x64), Windows 7 (x86 & x64), Windows Server 2012, Windows XP (x86) with Service Pack 3, Windows Vista (x86 & x64) with Service Pack 2, Windows Server 2003 (x86 & x64) with Service Pack 2 (verify MSXML6 is present), Windows Server 2003 R2 (x86 & x64), Windows Server 2008 (x86 & x64) with Service Pack 2, and Windows Server 2008 R2 (x64).

The amount of memory can be an important factor, especially if you intend to work on large images. A minimum of 512 MB of RAM is recommended, but the more RAM the better. Although ImageMagick runs well on a single core computer, it automatically runs in parallel on multi-core systems reducing run times considerably.

The Windows version of ImageMagick is self-installing. Simply click on the appropriate version below and it will launch itself and ask you a few installation questions. Versions with *Q8* in the name are 8 bits-per-pixel component (e.g. 8-bit red, 8-bit green, etc.), whereas, *Q16* in the filename are 16 bits-per-pixel component. A Q16 version permits you to read or write 16-bit images without losing precision but requires twice as much resources as the Q8 version. Versions with *dll* in the filename include ImageMagick libraries as [dynamic link libraries](#). Unless you have a Windows 32-bit OS, we recommend this version of ImageMagick for 64-bit Windows:

Version	HTTP	FTP	Description
ImageMagick-7.0.7-31-Q16-x64-dll.exe	<a href="#">download</a>	<a href="#">download</a>	Win64 dynamic at 16 bits-per-pixel component

Or choose from these alternate Windows binary distributions:

Version	HTTP	FTP	Description
ImageMagick-7.0.7-31-Q16-x64-static.exe	<a href="#">download</a>	<a href="#">download</a>	Win64 static at 16 bits-per-pixel component

# Преобразование типов файлов

Программа (утилита) `convert` может использоваться для преобразования форматов графических файлов.

- преобразование изображения формата `.png` в формат `.jpg`:

```
1 | convert myPhoto.png myPhoto.jpg
```

- при преобразовании в формат, предполагающий потерю качества (`.jpg`) можно указать степень сжатия:

```
1 | convert myPhoto.png -quality 10 myPhoto.jpg
```

для формата jpg это число обычно не ниже 75.

# Качество

Исходное изображение



102 Кб

Качество (сжатие) 10%



12 Кб

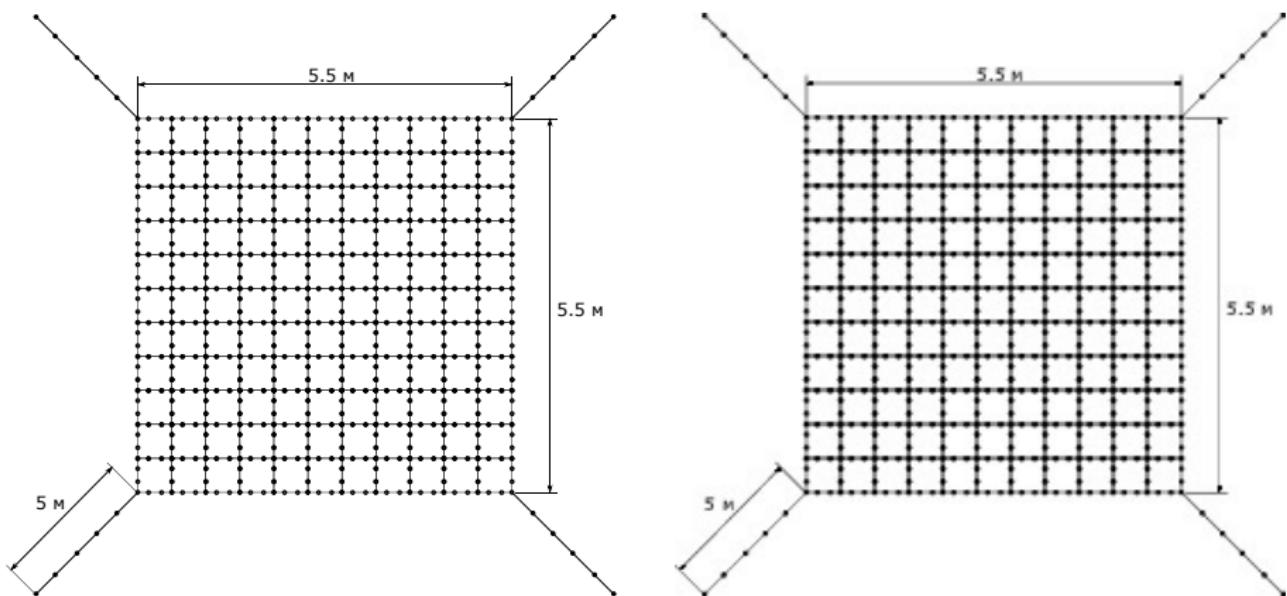
# Преобразование из векторных форматов

- Утилита `convert` может использоваться для преобразования векторных форматов (pdf, svg, eps, ps) в растровый.
- В этом случае указывается качество детализации получаемого растрового изображения в точках на дюйм при помощи свойства `-density`:
- В результате преобразования pdf в jpg

```
1 | convert -density 300 Document.pdf Document.jpg
```

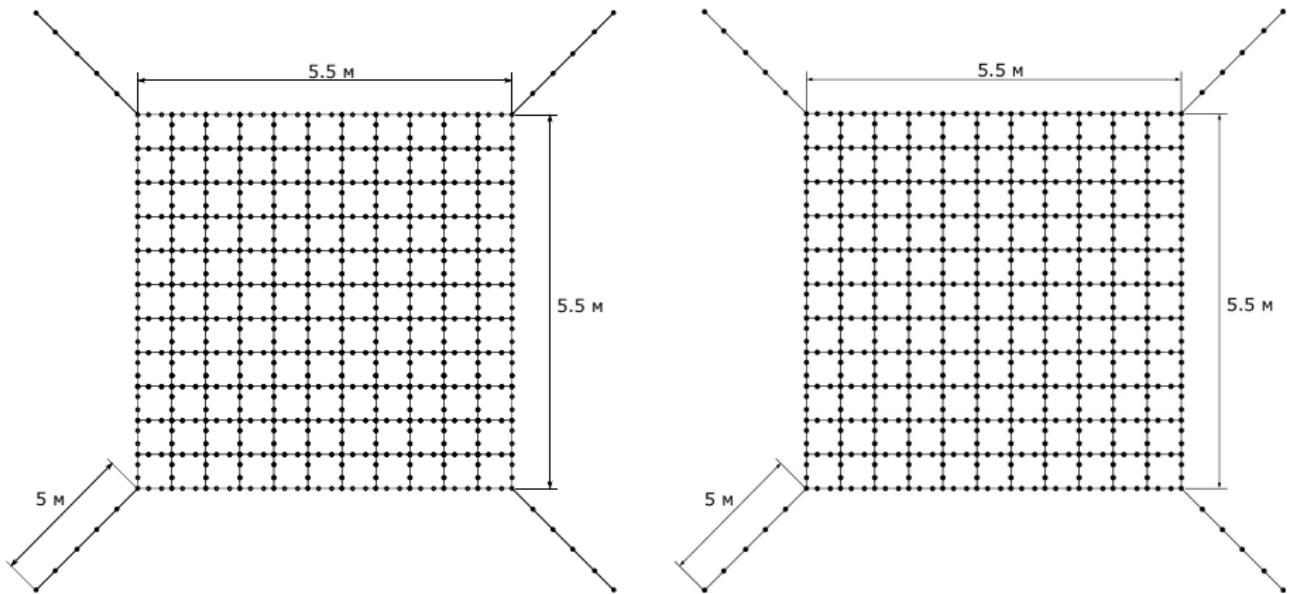
в каталоге появятся файлы Document-0.jpg, Document-1.jpg, ... Document-NN.jpg, где 'NN' – количество страниц в документе.

# Преобразование из векторных форматов



```
1 | convert Document.pdf -flatten -background white  
2 | Document.jpg
```

# Преобразование из векторных форматов



```
1 | convert -density 300 Document.pdf -flatten  
2 | -background white Document.jpg
```

## Извлечение одной или нескольких страниц

- Извлечение только первой страницы из документа `pdf`:

```
1 | convert -density 300 document.pdf[0] image.jpg
```

- Извлечение первых десяти страниц:

```
1 | convert -density 300 document.pdf[0-9] image.jpg
```

image-1.jpg, image-2.jpg, image-3.jpg, ... , image-9.jpg

- Извлечение первых десяти страниц (с указанием формата номера файлов-результатов):

```
1 | convert -density 300 doc.pdf[0-9] image_%02d.jpg
```

image\_01.jpg, image\_02.jpg, image\_03.jpg, ... , image\_09.jpg

# Преобразование формата

- Преобразование формата с перезаписью исходного файла выполняется при помощи программы `mogrify`

```
1 | mogrify -format jpg image.png
```

Файл `image.png` преобразуется в формат `jpg`. Исходный файл удаляется!

- Преобразование всех файлов формата `.png` в текущем каталоге в формат `.jpg`:

```
1 | mogrify -format jpg *.png
```

# Изображения в pdf

Программа `convert` может использоваться для преобразования растровых форматов, например, отсканированных страниц, в многостраничный формат pdf:

- Все изображения формата jpg в текущем каталоге преобразуются в многостраничный формат PDF

```
1 | convert *.jpg Document.pdf
```

# Размер изображения

- Уменьшение изображения на 50%:

```
1 | convert original.jpg -resize 50% small_50p.png
```

- Изменение размеров с указанием ширины и высоты  
(изображение вписывается в заданный прямоугольник):

```
1 | convert original.jpg -resize 500x800 img500x800.png
```

# Поворот

- Поворот на 90 градусов по часовой стрелке:

```
1 | convert -rotate "90" in.jpg out.jpg
```

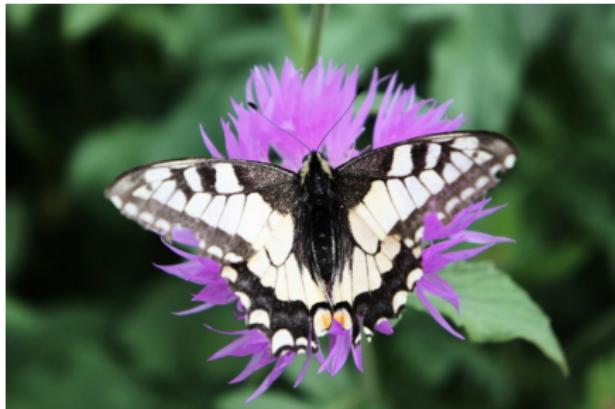
- Поворот на 45 градусов против часовой стрелки:

```
1 | convert -rotate "-45" in.jpg out.jpg
```

# Изменение яркости и контраста

Увеличение контраста:

```
1 | convert -contrast input.jpg output.jpg
```



Уменьшение контраста:

```
1 | convert +contrast input.jpg output.jpg
```

## Изменение яркости

Увеличение яркости (от исходного значения 100):

```
1 | convert input.jpg -modulate 150 output.jpg
```



Уменьшение яркости:

```
1 | convert input.jpg -modulate 50 output.jpg
```

## Изменение насыщенности

Увеличение насыщенности (второй параметр `-modulate`):

```
1 | convert input.jpg -modulate 130,150 output.jpg
```



Градации серого:

```
1 | convert input.jpg -modulate 50,0 output.jpg
```

# Изображение в градациях серого

```
1 | convert image1.jpg -colorspace gray image2.jpg
```



# Негатив

```
1 | convert image1.jpg -negate image2.jpg
```



# Сепия

```
1 | convert image1.jpg -sepia-tone 55% image2.jpg
```



## Кадрирование

Для обрезки изображения используется опция `-shave dx,dy`. Срезка слева и справа по 150 точек, снизу и сверху – 0:

```
1 | convert image1.jpg -shave 150x0 image2.jpg
```

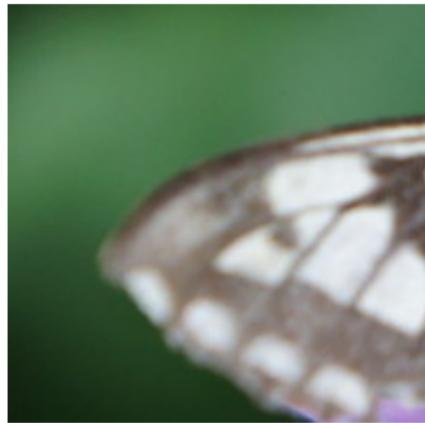
# Кадрирование

```
1 | convert image1.jpg -shave 150x20 image2.jpg
```



# Кадрирование

```
1 | convert FLY.jpg -crop 200x200+150+250 FLY_crop.jpg
```



Вырезать фрагмент 200 на 200 точек, смещённый на 150 точек по горизонтали и 250 по вертикали относительно верхнего левого угла (начало координат изображения).

## Аннотация

Подпись в левом верхнем углу (0,0) с отступом (полем) 10 по горизонтали и 10 по вертикали:

```
1 | convert image.jpg -annotate 0x0+10+10  
2 | "www.mypage.ru" image_annotated.jpg
```



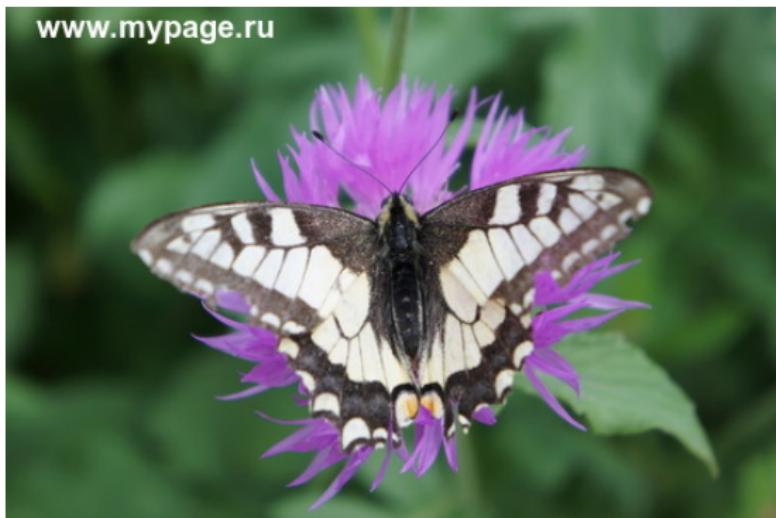
## Аннотация. Цвет текста

```
1 convert image.jpg -fill white  
2           -annotate 0x0+10+10 "www.mypage.ru"  
3           FLY_annotated.jpg
```



## Аннотация. Размер текста

```
1 convert image.jpg -fill white -pointsize 20  
2 -annotate 0x0+20+20 "www.mypage.ru"  
3 FLY_annotated.jpg
```



## Аннотация. Шрифт

```
1 convert image.jpg -fill white -pointsize 20  
2           -font "Comic Sans MS Bold.ttf"  
3           -annotate 0x0+20+20 "www.mypage.ru"  
4           FLY_annotated.jpg
```

Бабочка



# Создание gif анимации

- Даны последовательность изображений (файлов).
- Необходимо создать анимированный gif из этих изображений-кадров с заданным интервалом времени между кадрами.

```
1 | convert -delay 30 p1.png p2.png p3.png  
2 |         -loop 0 animation.gif
```

-delay 30 – интервал между кадрами 30 мс.

-loop 0 – количество повторений (0 бесконечный цикл).

- Все изображения по шаблону имени

```
1 | convert -delay 30 page_*.png -loop 0 animation.gif
```

# **FFMPEG**

# FFMPEG

Консольный видеоредактор (videoredactor командной строки), предназначенный для преобразования файлов. Основные утилиты:

- `ffmpeg` – программа для преобразования и редактирования видеофайлов
- `ffplay` – видеоплеер
- `ffprobe` – программа для извлечения информации о видеофайле

## Варианты использования

- Преобразование видеофайла из одного формата в другой (mp4-wmv, mp4-webm, ...)
- Изменение размера видеоизображения
- Изменение частоты кадров
- Монтаж (извлечение фрагментов видеопотока)
- Создание анимированного gif из видеофайла
- Добавление текста на видеоизображение

# Установка

<https://www.ffmpeg.org/>

The screenshot shows the FFmpeg download page at <https://www.ffmpeg.org/>. A red arrow points from the 'Download' link in the sidebar to the main content area. Another red arrow points from the 'Windows Packages' section in the bottom-left to the 'Windows Builds' button.

**FFmpeg**

About

News

Download

Documentation

Community

- Mailing Lists
- IRC
- Forums
- Bug Reports
- Wiki

Developers

- Source Code
- Contribute
- FATE
- Code Coverage

More

- Donate ❤
- Hire Developers
- Contact
- Security
- Legal

## Download FFmpeg

More downloading options

### Get the packages

Windows Packages

Windows Builds

More releases

Download  
ffmpeg-4.0.tar.bz2

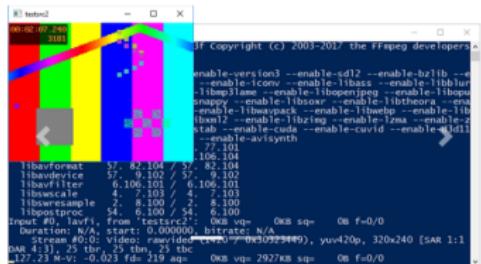
You can re...  
git clone  
Cannot ac...

# Установка

<https://ffmpeg.zeranoe.com/builds/>

## FFmpeg Builds

FFmpeg is the leading multimedia framework to decode, encode, transcode, mux, demux, stream, filter and play. All builds require at least Windows 7 or Mac OS X 10.10.



### Version

20180508-293a6e8

4.0

### Architecture

Windows 64-bit

Windows 32-bit

macOS 64-bit

### Linking

Static

Shared

Dev

Download Build

## Other Downloads

[Source Code](#)

[Readme](#)

# Формат запуска программы

```
1 | ffmpeg .1. .2. -i input_file .3. output_file
```

- .1. - глобальные опции
- .2. - опции для исходного файла
- .3. - опции для файла-результата

# Преобразование формата

Преобразование видеофайла в формат Windows Media Video (\*.wmv) для гарантированной работоспособности видеофайла в презентации PowerPoint:

```
1 | ffmpeg -i net.avi -b:v 4M -vcodec msmpeg4  
2 |           -acodec wma2 net.wmv
```

Без аудиодорожки

```
1 | ffmpeg -i net.avi -b:v 4M  
2 |           -vcodec msmpeg4 -an net.wmv
```

- **-b:v 4M** – битрейт видеопотока (Мб/с)
- **-vcodec msmpeg4** – тип сжатия видео (“codec”) Microsoft MPEG-4
- **-acodec wma2** – тип сжатия аудио Windows Media Audio

## Типы сжатия

- Качество видеофайла-результата определяется “битрейтом” и типом сжатия.
- Чем выше битрейт видеопотока (`-b:v`), тем выше качество результирующего файла.
- Список поддерживаемых типов сжатия:

```
1 | ffmpeg -codecs
```

ИЛИ

```
1 | ffmpeg -encoders
```

# Совместимость

Перекодирование для поддержки большинства устройств:

```
1 ffmpeg -i input.avi  
2     -c:v libx264 -crf 23 -profile:v baseline  
3     -level 3.0 -pix_fmt yuv420p  
4     -c:a aac -ac 2 -strict experimental -b:a 128k  
5     -movflags faststart output.mp4
```

[libx264](#) или MPEG-4 Part 10, Advanced Video Coding (MPEG-4 AVC) наиболее распространённый стандарт эффективного кодирования видео. Поддерживает разрешение до 8192 на 4320 точек.

## Изменение размера и качества сжатия

Для изменения размера изображения используется опция **-s**  
**ШИРИНАxВЫСОТА:**

```
1 | ffmpeg -i net.avi -s 320x240 net_320x240.mp4
```

Можно указать только один размер, а второй установить в **-1**. В этом случае неуказанный размер (-1) будет вычисляться:

```
1 | ffmpeg -i net.avi -s 320x240 net_320x240.mp4
```

# Фрагменты

Вырезать из видеофайла `input.wmv` фрагмент с 30 секунды продолжительностью 10 секунд:

```
1 | ffmpeg -ss 00:00:30.0 -i input.wmv  
2 |         -c copy -t 00:00:10.0 output.wmv
```

- `-ss 00:00:30` – с 30 секунды
- `-t 00:00:10.0` – продолжительность 10 секунд
- `-c copy` – копировать видео и аудиопотоки (без “пережатия”)

# Склейка

Для склейки нескольких видеофайлов, закодированных в одном формате, необходимо создать текстовый файл

[my-video.txt](#)

```
1 | file 'file1.avi'  
2 | file 'file2.avi'  
3 | file 'file3.avi'
```

---

```
1 | ffmpeg -f concat -i my-video.txt -c copy output.avi
```

## Извлечение одного кадра из видеофайла

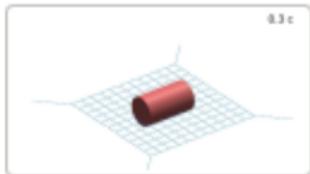
```
1 | ffmpeg -ss 00:00:01 -i video.avi -vframes 1  
2 |         video_frame.png
```

## Извлечение нескольких кадров из видеофайла

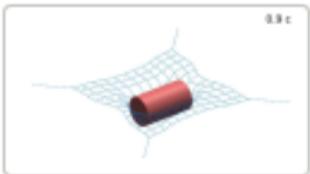
- Исходный видеофайл `net_case_1.avi`
- Необходимо получить изображения (“стоп-кадры”) с шагом 1 с.

# Извлечение кадров

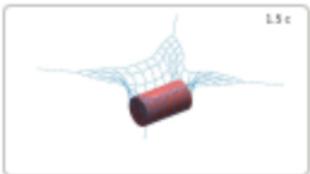
```
1 | ffmpeg -i net_case_1.avi -vf fps=1 case_1_%3d.png
```



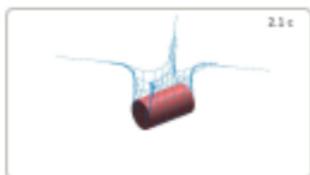
case\_1\_001.png



case\_1\_002.png



case\_1\_003.png



case\_1\_004.png



case\_1\_005.png



case\_1\_006.png



case\_1\_007.png



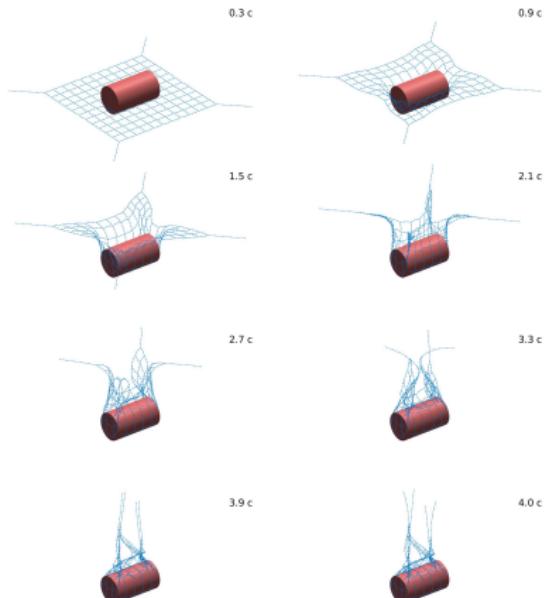
case\_1\_008.png



net\_case\_1.avi

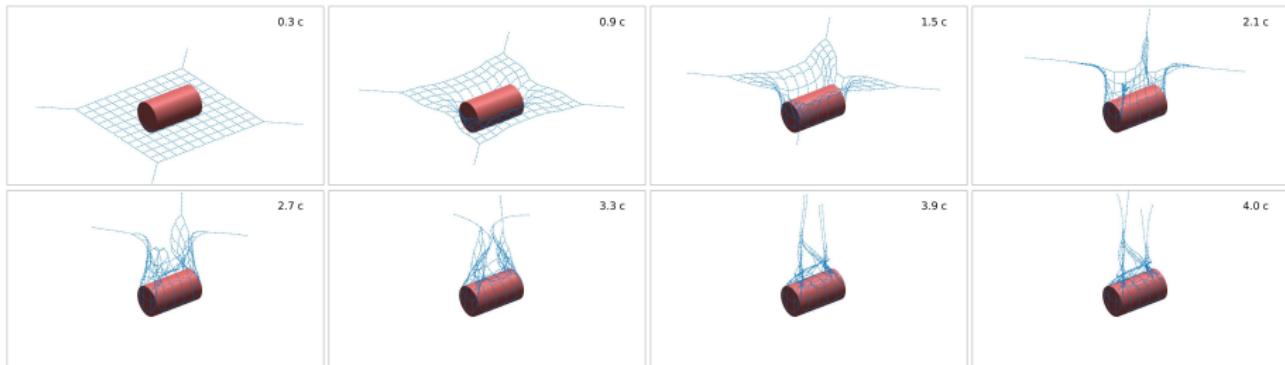
# Построение “галереи”

```
1 | montage case_1_*.png -tile 2x -geometry +1+1 gal1.png
```



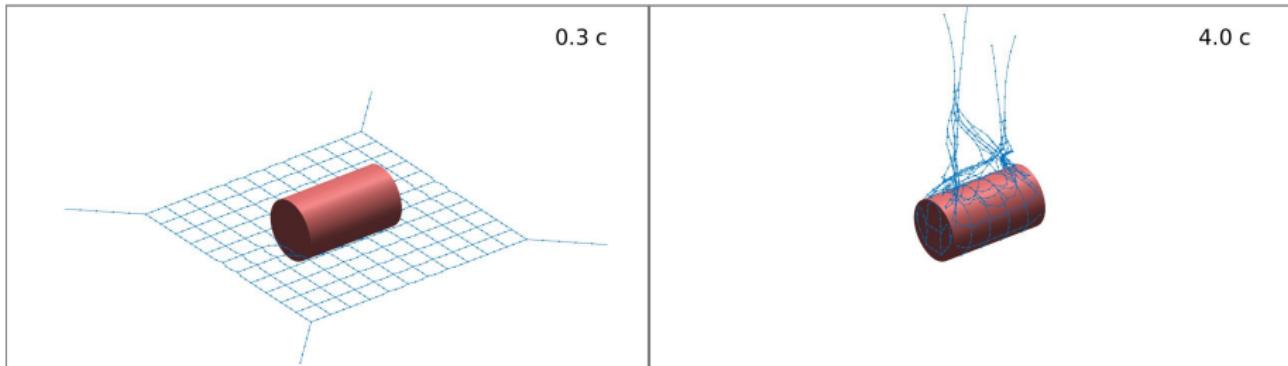
# Построение “галереи”

```
1 montage case_1_*.png -border 2x2  
2     -bordercolor gray -tile x2  
3     -geometry +10+10 gal2.png
```



# Построение “галереи” в рамках

```
1 montage case_1_001.png case_1_008.png  
2           -border 2x2 -bordercolor gray  
3           -tile 2x -geometry -2-2 gal3.png
```



# Видеопрезентация из изображений

- есть несколько изображений p01.png, p02.png, p03.png, ...
- требуется создать видеофайл продолжительностью 5 секунд
- каждые 5 секунд должно появляться новое изображение
- частота кадров видеофайла 30 кадров в секунду

```
1 | ffmpeg -r 1/5 -i p%2d.png -c:v libx264 -r 30  
2 |           -y -pix_fmt yuv420p slide_show.mp4
```

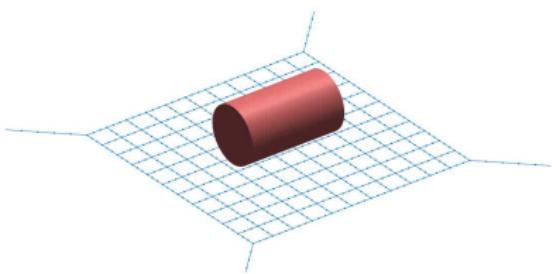
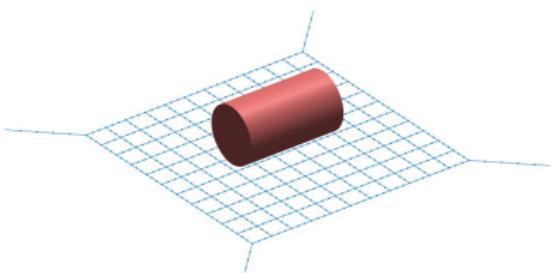
# Добавление аннотации

```
1 ffmpeg -i net.avi  
2     -vf "drawtext=text='Net capture ':  
3         fontcolor=black@0.6:fontsize=18:box=1:  
4             boxcolor=black@0.0:x=5:y=5"  
5     -b:v 4M -y net_annotated.avi
```

Захват сетью

0.0 с

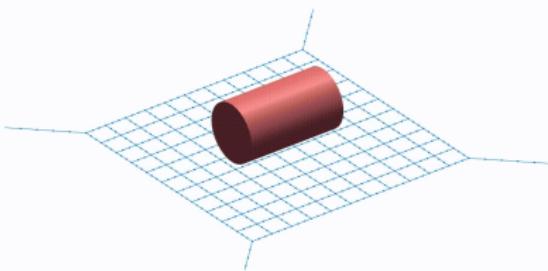
0.0 с



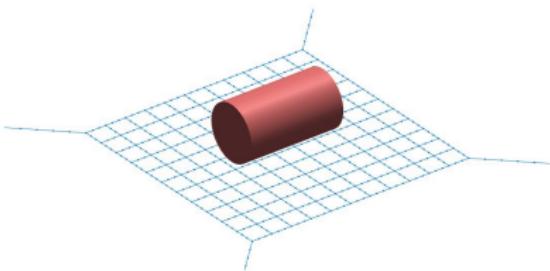
# Анимированный gif из видеофайла

```
1 | ffmpeg -i video.avi video.gif
```

0.0 c



0.0 c



## Задание

Используя анимацию движения механизма из курсовой работы по механике, построить галерею  $2 \times 3$  из шести характерных изображений положений механизма.

## Источники

- <https://www.imagemagick.org>
- [http://najomi.org/\\_nix/imagemagick](http://najomi.org/_nix/imagemagick)
- <https://www.ffmpeg.org/>
- <https://habrahabr.ru/post/333664/>