

# Image editor

[github.com/andy489](https://github.com/andy489)

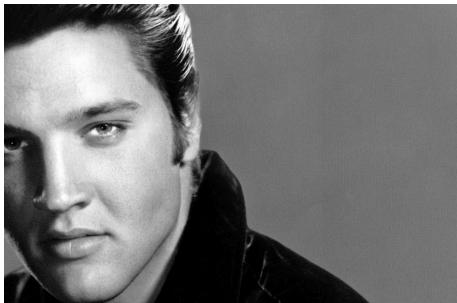
## План

Имаме директория, която съдържа снимки. Тези снимки са с най-различни размери. Дадено ни е някакво съотношение, в което трябва да изрежем всички снимки (за примерите ще използваме съотношение 3:2, но скриптовете, които пишем, трябва да работят с произволно). Най-простият начин да изрежем снимка е да отрежем прозорец в центъра, но това не винаги дава най-добрая резултат (например, следната картичка е по-красиво да я изрежем с прозорец, който е вляво, отколкото прозорец, който е в центъра):

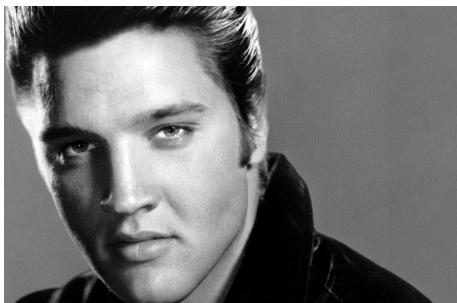
1. Оригиналното изображение



2. Прозорец в центъра



3. Прозорец вляво



Затова, за всяка снимка ще определим „интересна точка“, и ще се опитаме да генерираме прозорец, който е максимално голям, центърът му е максимално близо до интересната точка, и съотношението му е правилно.

## Помощни материали

На сървъра [astero.openfmi.net](https://astero.openfmi.net) са предоставени следните инструменти:

### скрипт *rw*

С него можем да преглеждаме изображения. Подаваме изображения като аргументи и скриптът връща на *stdout* линк, който можем да използваме, за да ги видим визуално:

```
$ pw pika.png  
https://astero.openfmi.net/preview/f563c5sbz1
```

Можем да подадем повече от един аргумент. Също, можем да *hover*-ваме с мишката над изображението, което ни показва координатите на пиксела под мишката. Тази възможност ще ни е удобна по-нататък.

### imagemagick toolkit

Той се състои от следните компоненти (и други, които няма да ползваме):

- командата *identify*, която връща информация за подадено изображение
- командата *convert*, която извършва манипулации върху подадено изображение и записва резултата в нов файл

Въсъщност, на нас ще ни трябват само две конструкции, които даваме тук, за да си спестим четене на документацията:

- *identify foo.png*, за да видим размера на изображението *foo.png*
- *convert foo.png -crop 200x300+10+50 +repage bar.png*, за да изрежем правоъгълник с размер 200x300 с горен ляв ъгъл на координати (10, 50) от изображението *foo.png* и да запишем резултата в ново изображение *bar.png*

Документация: <http://www.imagemagick.org/Usage/>

### команда *facedetect*

С нея можем да откриваме човешки лица в изображения.

Например:

```
$ facedetect elvis.jpg  
485 105 95 95  
816 138 84 84
```

*facedetect* извежда информация за всяко открито лице на отделен ред. Без допълнителни параметри, извежда 4 числа, които са координати на горен-лев ъгъл и размер на правоъгълника, описващ лицето.

На нас ще ни свърши работа по-прости вариант, *facedetect -c*, който извежда само координатите на центровете на откритите лица:

```
$ facedetect -c elvis.jpg
```

```
532 152
```

```
858 180
```



Документация: <https://www.thregr.org/~wavexx/software/facedetect/>

## Процес на работа

На сървъра има директория `/tmp/elvises`, в която има няколко примерни входни снимки. Може да ги използваме, за да тествате скриптовете си.

### Първа част

Ще напишем скрипт `crop.sh`, който при подаден първи аргумент `ipoint`, се извиква по следния начин:

```
$ crop.sh ipoint <a> <b> <x> <y> <input image> <output image>
```

Където:

- `<a>` и `<b>` дефинират съотношение на търсения размер
- `<x>` и `<y>` дефинират координати на интересна точка
- `<input image>` и `<output image>` са съответно входен и изходен файл

Например:

```
$ crop.sh ipoint 3 2 150 321 foo.jpg foo_cropped.jpg
```

Нашия скрипт трябва да изреже изображението във входния файл и да запише резултата в изходния файл, така че:

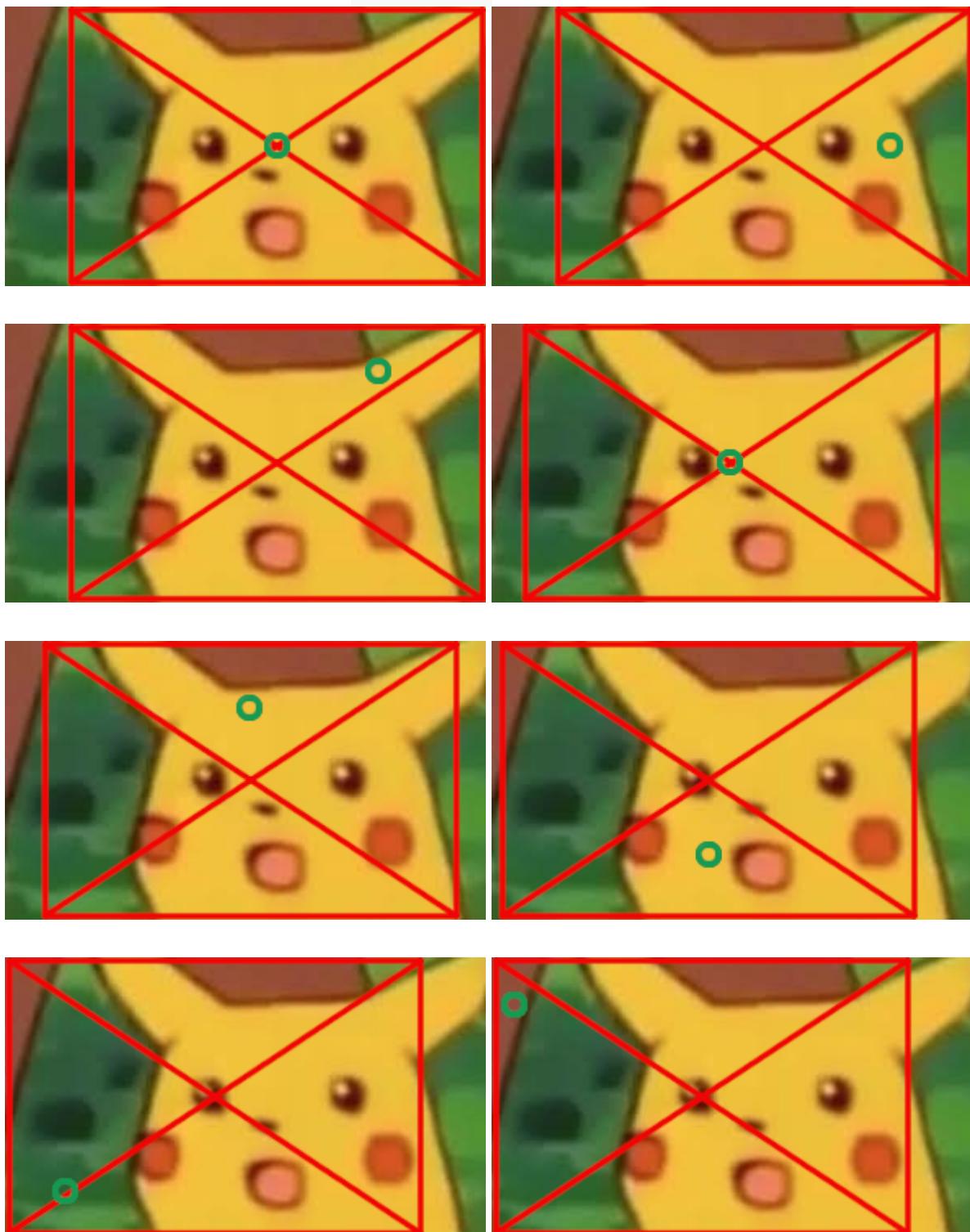
- размерът на изходния файл да има съотношение `a:b`, ако `a > b` и `sx > sy` или да има съотношение `b:a`, ако `a > b` и `sy > sx`, където входното изображение има размер `sx : sy`. С други думи, портретните снимки остават портретни, а пейзажните остават пейзажни. Ако е подадено `b > a`, може да ги размените или да изведете съобщение за грешка, по ваш избор.
- да няма начин да се изреже изображението, така че да получим резултат с по-голям размер, или да получим резултат, в който интересната точка е по-близо до центъра.

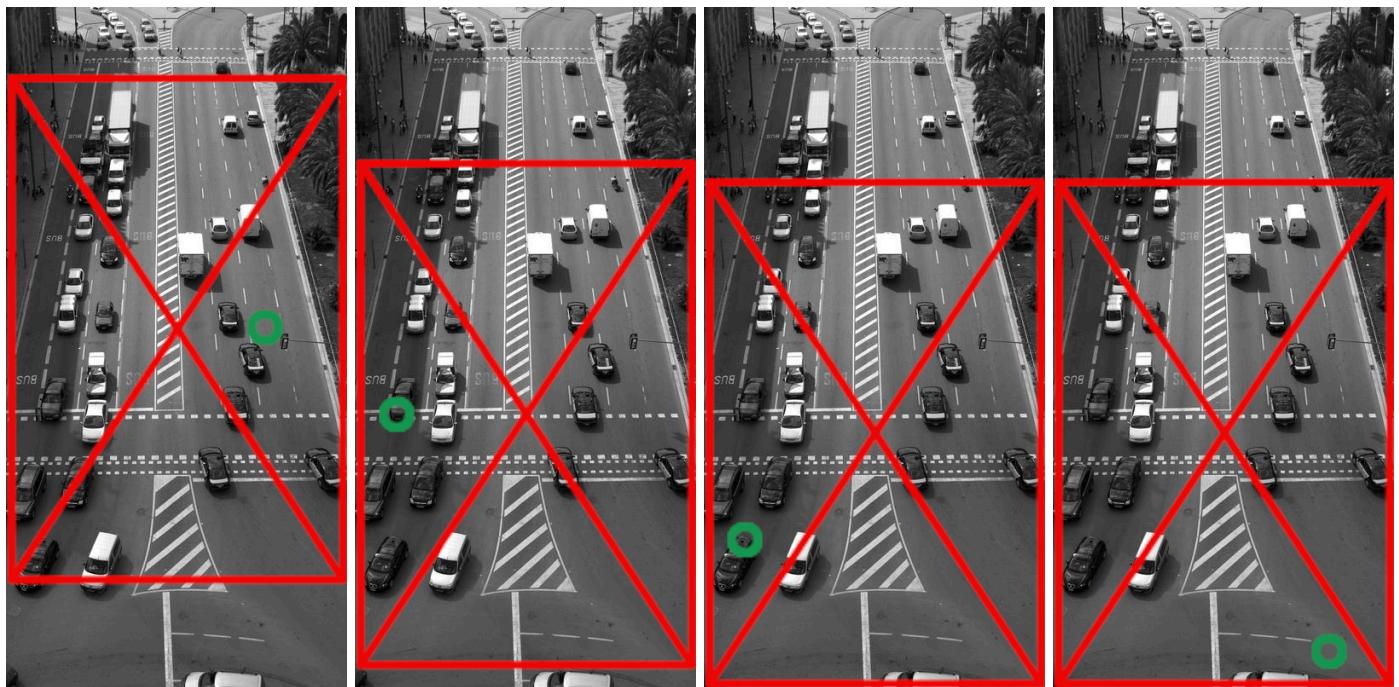
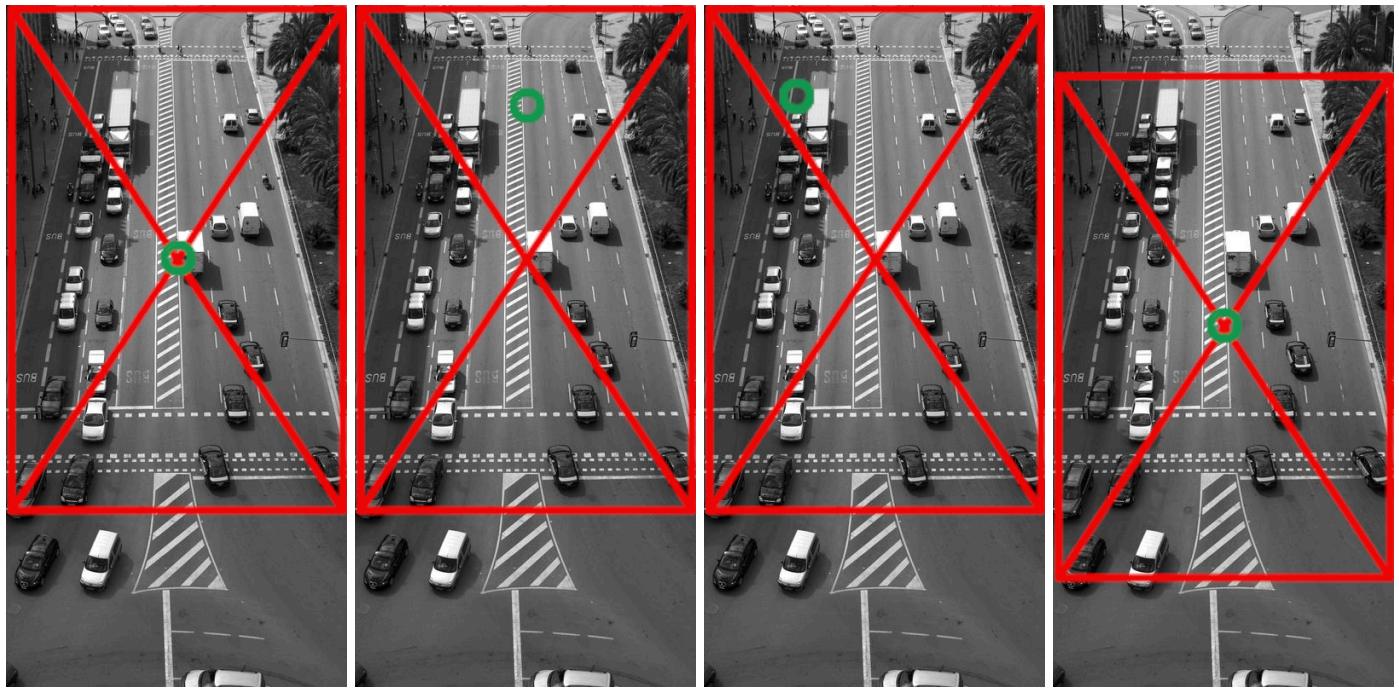
Тъй като това описание звучи прекалено сложно, вместо да го четем няколко пъти, за да го разберем, ще е по-лесно да погледнем следните примери:

## Примери

Тези примери показват всички възможни случаи, които трябва да бъдат обработени. На всеки от примерите имаме входното изображение, като върху него с зелено кръгче (○) е показана интересната точка, а с червена рамка (☒) е показана областта, която вашият скрипт трябва да изреже.

Във всеки от тези примери  $a:b = 3:2$ .





## Втора част

Ще допълним скрипта `crop.sh`, така че при подаден първи аргумент `faces`, той се извиква по следния начин:

```
$ crop.sh faces <a> <b> <input image> <output image>
```

и работи по същия начин както варианта с `ipoint`, само че вместо да приема интересната точка като аргумент, избира за интересна точка медианата на всички човешки лица в изображението. Ако няма човешки лица, скриптът избира центъра на изображението за интересна точка.

*Bonus: ще използваме друг евристичен метод за избор на интересна точка, различен от този на медианата - центъра на най-малкия правоъгълник, който покрива множеството от всички намерени човешки лица.*

*В този случай скрипта crop.sh ще се извиква по същия начин както е показано по горе с тази разлика, че аргумента ще бъде faces\_r*

### Трета част

Ще напишем скрипт скрипт crop\_all.sh, който се извиква по следния начин:

```
$ crop_all.sh <a> <b> <in_dir> <out_dir>
```

Където <a> и <b> пак дефинират съотношение, <in\_dir> е директория, в която има много на брой изображения, а <out\_dir> е директория, в която скриптът трябва да записва резултатните изображения.

Скриптът ще минава през всяко изображение, и да показва на потребителя избор между три евристики, така че той лесно да може да види резултатните изображения и да си избере:

- Център на изображението
- Медианата на човешките лица
- Ръчно въвеждане на координати на интересна точка

[github.com/andy489](https://github.com/andy489)