How do I create an animated gif from still images (preferably with the command line)?

Asked 8 years, 11 months ago Modified 6 months ago Viewed 230k times

177

I want make a `.gif` animated picture from a given set of `.jpg` pictures.

I would prefer to do it from the command line, so command line tools would be very welcome.

command-line image-processing gif

Share Improve this question

Follow

edited Feb 3, 2019 at 17:39

pomsky

68.8k 21 240 249

asked Jul 14, 2015 at 10:57

Maythux

84.7k 54 244 275

3 Works, very memory hungry though. There are LOADs of potential dupes though:askubuntu.com/questions/648244/… askubuntu.com/questions/457351/… askubuntu.com/q/573712/178596 askubuntu.com/questions/566476/… askubuntu.com/questions/380875/… askubuntu.com/q/636149/178596 – Wilf Jul 14, 2015 at 11:48

9 Answers

Sorted by: Highest score (default)

198

You can use ImageMagick package. Install it using the command:

```
sudo apt-get install imagemagick
```

Now you can create a `gif` from number of pictures(`jpg`) using:

```
convert -delay 20 -loop 0 *.jpg myimage.gif
```

Share Improve this answer

Follow

edited Jul 22, 2015 at 7:32

answered Jul 14, 2015 at 10:57

Maythux

84.7k 54 244 275

12 Please, include here how you resize the gif animation etc by 50%. Etc `-resize 50%` .
– Léo Léopold Hertz 준영 Aug 7, 2016 at 11:23

3 Ubuntu 18.10 failed with "cache resources exhausted" for these images: github.com/cirosantilli/media/blob/master/… related: superuser.com/questions/1178666/… – Ciro Santilli OurBigBook.com Dec 23, 2018 at 16:40

ImageMagick users will likely also want: `-deconstruct` . I have also found that ImageMagick is very memory hungry, see benchmarks: askubuntu.com/a/1102183/52975
– Ciro Santilli OurBigBook.com Dec 23, 2018 at 17:48

`convert -delay 0 -loop 0 *.tif myanimated.gif` worked nicely, on 720 tif files – ron Jul 17, 2019 at 15:34

awesome, you may change the delay according to your needs. – sh6210 Sep 26, 2021 at 8:24

To complete @Maythux answer:

83

To avoid generating a very large file, you can use `-resize` option:

In my case, I have 4608x3456 images and the generated gif was more than 300M for 32 images

```
convert -resize 20% -delay 20 -loop 0 *.jpg myimage.gif
```

or

```
convert -resize 768x576 -delay 20 -loop 0 *.jpg myimage.gif
```

Take care of *.jpg

`*.jpg` sucks a bit when dealing with numeric values, you may generate a gif with unsorted pics.

```
$ ls|cat
21-33-26_1.jpg
21-33-26_10.jpg   // <--- this one
21-33-26_2.jpg
21-33-26_3.jpg
21-33-26_4.jpg
21-33-26_5.jpg
21-33-26_6.jpg
21-33-26_7.jpg
21-33-26_8.jpg
21-33-26_9.jpg
21-33-28_1.jpg    // <--- should be here
21-33-28_2.jpg
21-33-28_3.jpg
...
```

As the shots were taken very quickly (10/s) they all have the same modification time and you can't trick using `ls -t` for example. On ubuntu you can use `ls -v` instead, something like:

```
convert -resize 768x576 -delay 20 -loop 0 `ls -v` myimage.gif
```

Sorting numerically is quite tricky on Mac OS X though, I guess you'll need to build a custom script.

Share Improve this answer Follow

answered Apr 16, 2016 at 9:28

Alain

8 You can avoid your *.jpg issue by forward padding numbers with zeros. "01.jpg" instead of "1.jpg", and so on. If you get to triple digits, then "001.jpg", "010.jpg", etc. – bigreddmachine Nov 28, 2016 at 23:53

2 There are several ways around the filename sequence problem. Including `find`, `sort`, brace expansion, and so on. The `ls` tool is notoriously unsuitable for this kind of thing. Use `find`. There's a bit of a learning curve, but it's worth it. – voices Jan 23, 2017 at 16:28

Some users might be interested in editing filenames with massren: github.com/laurent22/massren – Graham P Heath Nov 30, 2017 at 22:09

it takes ages to finish. Is there a verbose option? – ouranos Jul 18, 2019 at 16:19

2 If the input has a transparent background, I would also recommend the `-dispose Background` option. (I realize OP's post is about jpeg images which cannot have transparent backgrounds, but just in case anyone finds this answer via Google, like I did). By the way, regarding the `ls` sorting, you could always do some bash script hacking inline, e.g.: `$(for f in `seq 0 10`; do printf "21-33-26_$f.jpg\n"; done)` – Daniel Oct 31, 2022 at 21:46

55

ffmeg solution + test data

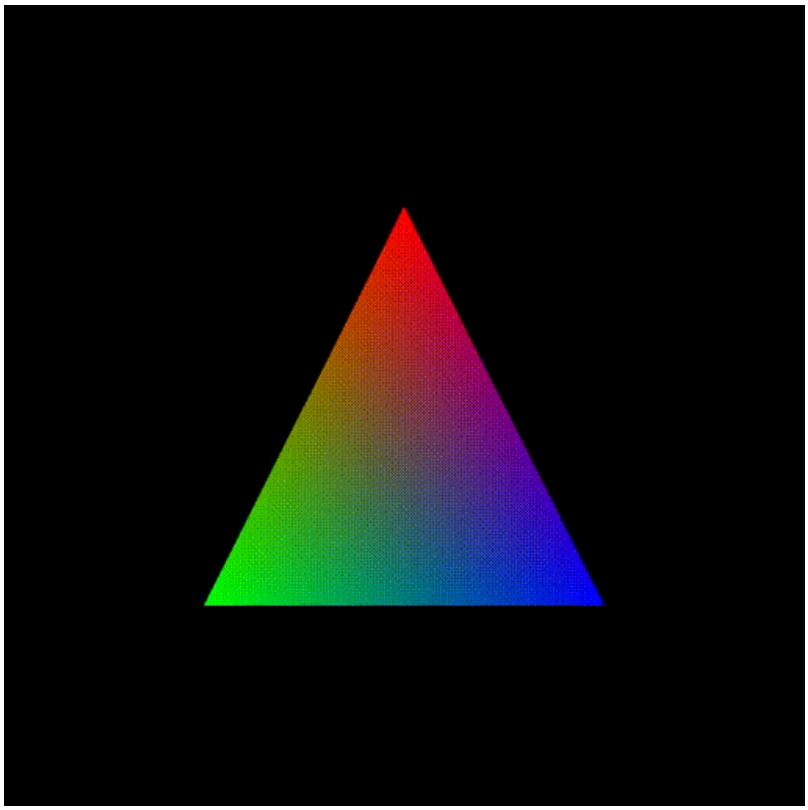
As of Ubuntu 18.10, ffpmeg 4.0.2-2, ImageMagick 6.9.10-8, I have found that ffmpeg is much faster than ImageMagick, and uses much less memory.

The simplest conversion command is:

```
ffmpeg \
  -framerate 60 \
  -pattern_type glob \
  -i '*.png' \
  -r 15 \
  -vf scale=512:-1 \
  out.gif \
;
```

You can get my test data with:

```
wget -O opengl-rotating-triangle.zip https://github.com/cirosantilli/media
/blob/master/opengl-rotating-triangle.zip?raw=true
unzip opengl-rotating-triangle.zip
cd opengl-rotating-triangle
```



The test data was generated with: https://stackoverflow.com/questions/3191978/how-to-use-glut-opengl-to-render-to-a-file/14324292#14324292 and contains 256 1024x1024 PNG images.

The important `ffmpeg` options I want to highlight are:

- `-pattern_type glob` : convenient way to select images
- `-framerate 60` : assume 60 FPS on input images, and output the same FPS.

  `ffmpeg` cannot know otherwise, since there is no FPS data is in images as there is is in video formats.

  The 256 input frames take about 4 seconds to finish.

  `-r 15` : optional. Pick one every 4 images so reduce size (`4 == 60 / 15`).

  With it, `identify out.gif` says that the GIF contains only 64 frames.

  It still takes 4 seconds to play, so the delay is altered to make things match.
- `-vf scale=512:-1` : optional. Set the width, scale height proportionally, usually to reduce size and save space.

See also:

- video from images: https://stackoverflow.com/questions/24961127/how-to-create-a-video-from-images-with-ffmpeg/37478183#37478183
- GIF from video: How to create an animated GIF from MP4 video via command line?

ImageMagick vs ffmpeg benchmark

;

The commands were constructed to produce outputs that are as close as possible to make the comparison valid:

- `/usr/bin/time -v` : used to find the maximum memory usage as explained at: https://stackoverflow.com/questions/774556/peak-memory-usage-of-a-linux-unix-process
- `-deconstruct` : GIF images can contain just the minimal modified rectangle from the previous frame to make the GIF smaller.

  `ffmpeg` calculates those diffs by default, but ImageMagick does not, unless `-deconstruct` is used.

  You will basically want to use that option every time with ImageMagick.

  We can observe the difference with:

  ```
  identify out.gif
  ```

  With the compressed version, all frames have smaller sizes than the initial one, e.g.:

  ```
  out.gif[0] GIF 1024x1024 1024x1024+0+0 8-bit sRGB 256c 16.7865MiB 0.010u
  0:00.010
  out.gif[1] GIF 516x516 1024x1024+252+257 8-bit sRGB 256c 16.7865MiB 0.010u
  0:00.010
  out.gif[2] GIF 515x520 1024x1024+248+257 8-bit sRGB 256c 16.7865MiB 0.010u
  0:00.010
  ```

  In this example, the second frame is only `516x516` instead of the full 1024x1024, and is placed at an offset of `252+257` . It therefore contains just the middle triangle.

  See also: how can I resize an animated GIF file using ImageMagick?

- `-delay` : value that matches the 60FPS of `ffmpeg` . Should not matter for conversion performance, but I don't want to risk it.

The output GIFs have about the same size and look visually identical.

Test hardware: Lenovo ThinkPad P51 laptop, Intel Core i7-7820HQ, 32GB(16+16) DDR4 2400MHz SODIMM, 512GB SSD PCIe TLC OPAL2.

Share Improve this answer

Follow

edited Sep 16, 2020 at 6:16

answered Dec 15, 2018 at 20:52

Ciro Santilli OurBigBook.com

29k 14 119 113

2 Upvoted and liked!, great answer, I have one small follow up question. I have not so much high quality images, and i did not use `-r` and `-v` commands, still the output gif is of much lower quality than original png files. How to keep the same quality of gif as given png files? – BhishanPoudel Apr 10, 2019 at 2:27

Hi @astro123 how is the quality lower? Smaller image size, or less frames, or something else (GIF uses lossless apparently, so can't be worse compression)? Try `-framerate 60 -r 60` . Inspect the generated GIF and input images with ImageMagick tools like `identify` to try and figure out what is going on. – Ciro Santilli OurBigBook.com Apr 10, 2019 at 8:32

1 I would strongly recommend using `palettegen` and `paletteuse` to get much better quality and lower file size, e.g. with `-filter_complex "[0:v] split [a][b];[a] palettegen [p];[b][p] paletteuse"` . See: giphy engineering. Also, note that ffmpeg does not support transparent backgrounds, which is quite important for making animated emojis etc. – Daniel Oct 31, 2022 at 21:39

Instead of modifying file names you can use globbing to get your shell to expand file names

30

```
convert -resize 50% -delay 10 -loop 0 image_{0..99}.jpg output.gif
```

Share Improve this answer

Follow

edited Aug 3, 2021 at 7:15

Ciro Santilli OurBigBook.com

29k 14 119 113

answered Jul 27, 2017 at 18:52

shanksk

401 4 3

20

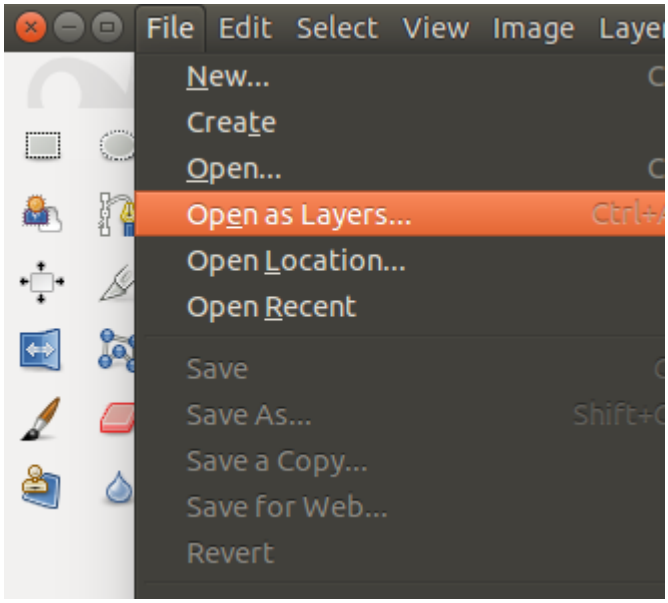
GIMP

You can easily do this with GIMP. First install it if it's not installed already with

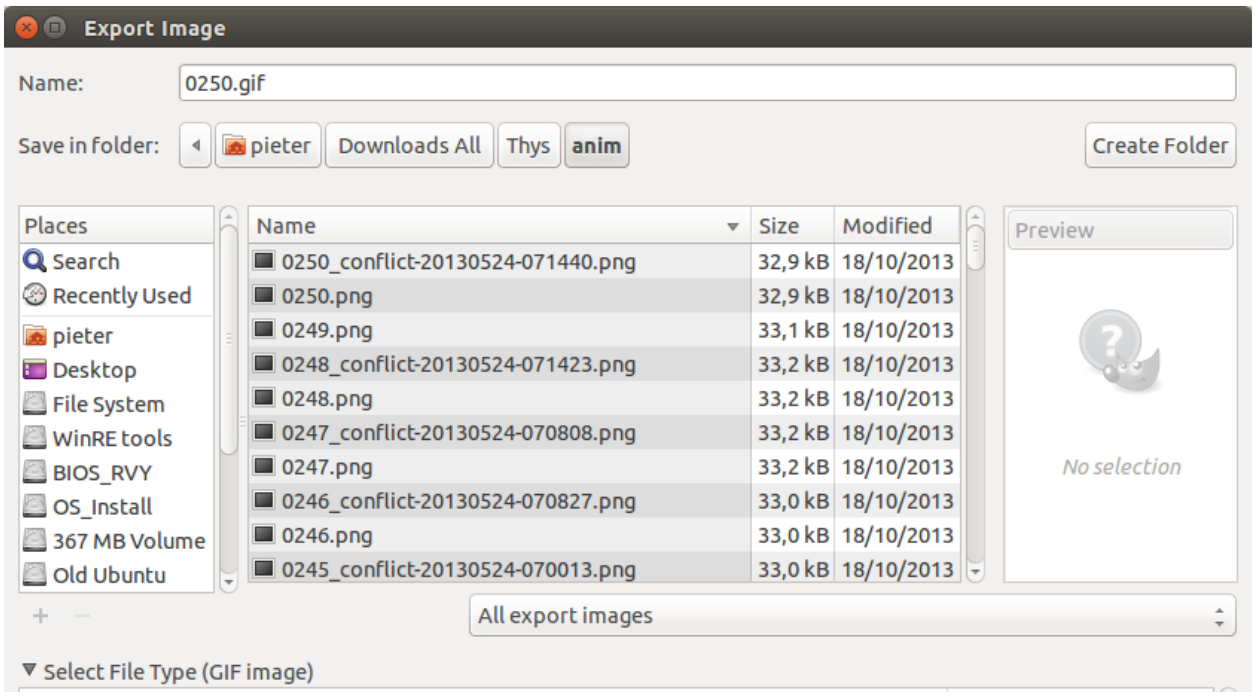
```
sudo apt-get install gimp
```

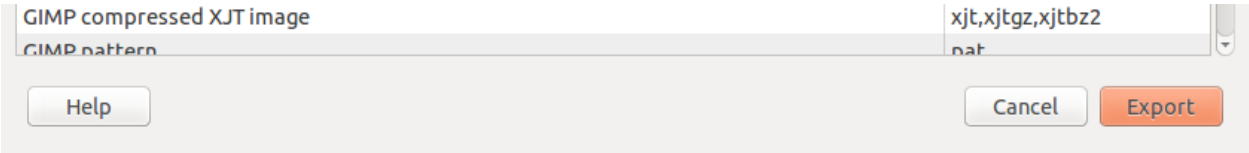
Creating the gif

From GIMP go to **File** -> **Open as Layers** to open all the png's on their own layer.

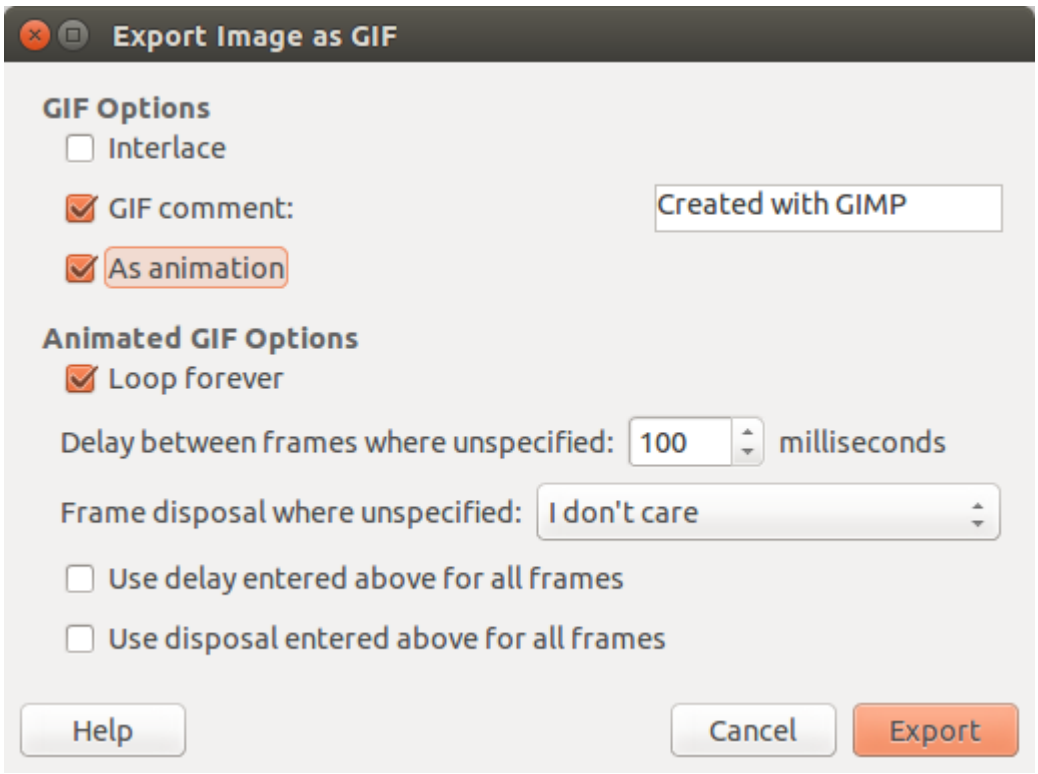


From here you can perform edits on the layers and, once done, go to **File** -> **Export As**. From the dialog be sure to set the file type to GIF.



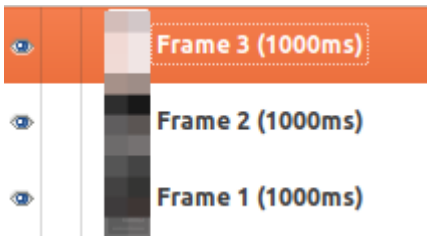


From there you will go to the GIF export options. Tick the '*As Animation*' option and set the parameters as required.



To change the delay between frames

Modify the name of the layers, and include the delay in milliseconds between parenthesises, like this: `(1500ms)`



To preview the animation before exporting

1 'Command line preferred' - from the question – Gathide Jan 8, 2022 at 19:44

5

You can use a program called convert included in the imagemagick package. It is command line driven, but very easy to use. Install it either through the software center, or go to a command prompt and type

```
sudo apt-get install imagemagick
```

Now to create the .gif.

```
convert -delay 100  -size 100x100 xc:SkyBlue \
          -page +5+10  balloon.gif   -page +35+30 medical.gif  \
          -page +62+50 present.gif   -page +10+55 shading.gif  \
          -loop 0  animation.gif
```

*Note the above example is straight from Image Magick Examples

Share Improve this answer

Follow

edited Feb 3, 2019 at 16:03

pomsky

68.8k 21 240 249

answered Apr 29, 2014 at 0:15

Scott Goodgame

2,636 15 20

1 How do I do 40 PNG images in a looping GIF? All in folder /home/fusion809/Documents/Images – Josh Pinto Apr 29, 2014 at 0:55

3 @BH2017: `convert -delay 20 -loop 0 *.png out.gif` makes a looping GIF. – unutbu Dec 24, 2018 at 0:22

1

To add something, you'll probably end with a huge GIF file with the `convert` variant proposed.

As this answer in StackOverflow states, it's convenient to optimize the resulting GIF with something like the following:

```
mogrify -layers 'optimize' -fuzz 7% mygif.gif
```

With this I'm getting about 1/50 times the original size.

I've been getting trouble with the resources assigned to `ImageMagick`. This post can be helpful if it's the case.

Share Improve this answer

Follow

edited Oct 27, 2022 at 21:00

answered Oct 27, 2022 at 20:58

Alfredo Tostón

111 5

1

just use **ffmpeg** on the command line, it comes preinstalled on Ubuntu, Kubuntu, Debian and other linux

```
ffmpeg -f concat -i list.txt output.gif
```

your *list.txt* file should be arranged like this:

```
image001.jpg
duration 5
image002.jpg
duration 12
image003.jpg
duration 7
image004.jpg
duration 2
```

Share Improve this answer Follow

answered Nov 25, 2023 at 1:53

user12711
245 2 9

1 This is better than most solutions, since it gives flexibility in terms of durations for each frame. But lines should rather look like this: file 'image001.jpg' – Genom Feb 25 at 11:06

1 I like this approach! In my case, I *do* have frames with different durations, so I was considering to discard `ffmpeg` and just tweak with the endless options available with ImageMagick. Now I will have to give it a try with `ffmpeg` ! Thanks 🙏 – Gwyneth Llewelyn Apr 10 at 18:04

0

I suggest you use the same convert command, but make sure to include `-dispose Background` to clear the canvas before adding the next frame. It looks like the following:

```
convert -delay 100 -dispose Background -loop 0 *.jpg animation.gif
```

Share Improve this answer Follow

answered Oct 13, 2023 at 22:18

Mohamed Ismaiel Ahmed
113 4