Note of usage to some command in imagemagick

October 22, 2017

Contents

1   Convert

1.1   Change format

This is too simple, and you can just execute:

$$\boxed{\text{convert } \textit{source target}}$$

and convert will change format of image by it's suffix.

But it is not good at to convert bitmap to vector, if you want to do it, you should have potrace installed and you can convert the source image to bmp, bpm, or pbm, and then type:

$$\boxed{\text{potrace } source \text{ -s -o } target}$$

'-s' option is for svg format, there are some other options for other vector formats, but unfortunately, potrace can only convert graphics with gray scale.

1.2   Cut

$$\boxed{\text{convert } source \text{ -crop } wxh \text{ } target}$$

This command was used to devide the source image into several subfigures with specified size, if the size of source was not the interger multiple of subfigures, and the size of subfigures in right and bottom will take the size in reality.

$$\boxed{\text{convert } source \text{ -crop } wxh+dx+dy \text{ } target}$$

'w' was the width of target photo and the 'h' was the height.
'dx' was the offset in the x axis and 'dy' was the offset in the y axis, the origin was the top left.

1.3   Make a color block file

$$\boxed{\text{convert -size } wxh \text{ xc:}color \text{ } target}$$

This command can also create a image with specified color and specified size.

$$\boxed{\text{convert -size } wxh \text{ xc:}color \text{ } source \text{ -geometry } +dx+dy \text{ -composite } target}$$

'color' was the color of background, you can use the name of color like 'cyan', 'blue', 'red', or the color code like '#44ff88'.

1.4 Replace color in images

convert *source* -fuzz *percent%* -fill *color1* -opaque *color2 target*

'percent' was the threshold value, and color2 was the color which will be replaced and the color2 was the color which will replace color1.

convert -negate *source target*

This command was used to reverse the color in the image.

If you want to reverse the color in only a part of image, you can use this:

convert -region *wxh+dx+dy* -negate *source target*

or

convert -gravity *direction* -region *wxh+dx+dy* -negate *source target*

And if argument 'direction' was chosen as center, the latter was located from where the area was in the center of source, you can use other direction like southeast and so on.

Also, you can use '-region' or '-gravity' to make other processing on only a part.

Some times, you don't know the code of color which you want to replace, and this command will help you:

convert *source* -crop *wxh+dx+dy* txt:-

This command will output the color code of all points in the area you pick.

1.5 Adjust some attribute of the whole image

convert *source* -type Grayscale *target*

This command will help you to convert a image to gray mode.

> convert *source* -bright-contrast *dbxdc target*

db is the increment of brightness, and the dc is the increment of contrast, this will adjust the brightness and contrast of the image.

1.6   Merge

> convert ± append *source1 source2 ... sourcen target*

The '±' decided the direction of merge, when it was '−', merge was in portrait, and when it was '+', merge was in landscape. You can put '-append' just behind the target, it's ok.

You can also use the background to merge these figures.

1.7   Border add

> convert *source* -bordercolor *color* -border *width target*

This command was used to add the border to the image in specified color and specified width.

You can also use this:

> convert -mattecolor "*color*" -frame *wxh source target*

This time 'w' was width of vertical border and 'h' was width of aclinic border.

You can use this to make the border solid:

> convert -mattecolor "*color*" -frame *wxh+down+up source target*

This was a command to remove border:

> convert *source* -trim *target*

1.8 Draw on the image

convert *source* -font *fontname* -fill *color* -pointsize *size* -annotate $+dx+dy$ *@textfile target*

'fontname' was the absolute path to the font file, the 'size was the size of words, 'dx' and 'dy' was offset, and 'textfile' was the file including the words, or you can directly replace '@textfile' by the words.

convert *source* fill *color1* -stroke *color2* -strokewidth *width* -draw *shape* $x1, y1x2, y2$ *target*

'color1' was the padding color filling the shape, and 'color2' was color of the line, width was linewidth, shape was the shape you want to draw on the image, such as rectangle, line, circle. x1,y1 was the origin coordinate and x2,y2 was target coordinate.

If you want to add mark to the specified graph, you can also use composite:

convert *source mark* -gravity *direction* -geometry +dx+dy -composite *target*

'mark' was the photo you want to add as a mark.

1.9 Zooming and rotating

convert -resize *w*x*h* *source target*

Because of zoom in equal ratio, so the target usually not in the size you specified, but at least one of width and height will match your request.
You can also only specified only one of the size, but if you specified height, remember to add 'x' before the height.

If you do not need ratio, you can add '!':

convert -resize *w*x*h*! *source target*

If you don't know whether the size of photo match your request, you can use '<' and '>' to control it:

$$\boxed{\text{convert -resize } "w\text{x}h \gtrless " \text{ source target}}$$

If you use '>', it will zoom when the size of image larger than specified; Or if '<', it will zoom when size smaller.

You can also use '!' to confirm the size behind '$\gtrless$'.

Rotate the source image in a certain angle:

$$\boxed{\text{convert -rotate } angle\% \text{ source target}}$$

1.10 Compress

$$\boxed{\text{convert -strip -quality } percent\ source\ target}$$

Quality was an attribute of image. the higher quality is, the bigger space it occupied. Usually, if you reduce the quality to 85, you can't find obvious difference, and the space occupied will reduce a lot.

And the profile recorded some information of image, such as the information of camera, the unit data in photoshop and the table of colors, you can use '-strip' to remove this, or you can also use '-profile "*"'.

1.11 Make animation gif

$$\boxed{\text{convert -delay } seconds\ source1\ source2\ \dots\ sourcen\ target}$$

'seconds' was the seconds delay between frames.

You can use '-loop *times*' argument to limit the times target play, and the default setting was 0, which represent loop with no limit.

And if you want to add a background to the gif:

$$\boxed{\text{convert -loop } times\ bgimage \text{ -page } +dx+dy\ source\ target}$$

If you want to devide a gif into frames, you can use this:

$$\boxed{\text{convert -coalesce } source\ target}$$

This will create several images and they will be named as target-0, target-1, . . . , target-n, as the frames in the gif.

1.12   Verbose

You can use identify to get the verbose information of image, but convert can also realize this function.

$$\boxed{\text{convert } \textit{source} \text{ -verbose } \textit{target}}$$

This command was used to output identification information of source image when you convert it into another format.

You can also convert the image into ASCII text file to get information in detail, it will point out the color of every point in the image.