NAME

ezthumb − generate thumbnails of video clips

SYNOPSIS

ezthumb [*options*] *video_clips...*

DESCRIPTION

ezthumb is used to create thumbnails for video files. It uses the FFMPEG library as its engine so **ezthumb** almost supports any video formats. Normally the screenshots would be tiled into a single canvas picture or compose a single **GIF animation** file. However it can also save frames separately. The output format could be **JPG, PNG, GIF** or **GIF animation** , where **PNG** and **GIF** supports the transparent background. Batch processing allows.

OPTIONS

−b, **−−bind**

Bind multiple video sources to generate thumbnails as one. All video sources must be listed in the command line. Note that the binding mode would not work with the **−R** recursive option.

−c, **−−colour** *MI:TM:BG*

Specify the colour of texts and canvas. *MI* is the colour of the media information. *TM* is the colour of the inset timestamp and *BG* is the canvas' background colour. These colour codes use the same form of **RRGGBB** which is composed by Red, Green and Blue. The **RRGGBB** must be set by 6 hexadecimal number to make sure the colour won't go wrong, for example: *0f55f0*.

−d, **−−during** *head*

Specify the method to retrieve the duration of videos. Normally the duration can be retrieved from the head of the video container, which is the **'head'** mode. However, some dodgy files were broken or included the inaccurate duration value. As an alternative the duration could be retrieved by **'scan'** mode. **ezthumb** would scan the whole video file to locate the last packet. It is accurate but slow. The compromise mode is **'fast' ,** which only scans the tail part of the video. The default is **'auto'** mode, which would use the **'head'** mode first. If the result is apparently too ridiculus, it will change to **'fast'** or **'scan'** according to the video attribute.

If **ezthumb** took less screenshots than expected, you may think about using the **'fast'** or **'scan'** options.

−f, **−−font** *filename*

Specify the full path of a TrueType font file, which would be used to render the media information and the inset timestamp in the output picture. If this option is not specified, the program will use the internal fix-pixel fonts. Note that the fontconfig patterns is also acceptable. For example: Utopia:Bold. See fc-list(1) for details. **ezthumb** uses UTF-8 for file names so practically it can support different languages if a proper font was choosen.

−F, **−−fontsize** *MI:TM*

Set up the pointsize of the font. **MI** is for the text size of the media information and **TM** is for the size of time stamps inside the screen shots.

−g, **−−grid** *column x row*

Specify the grid of the screenshots in the canvas. The default value is *4x4* which means to create a 4 columns by 4 rows thumbnail canvas. Therefore it would take 16 screen shots in total. The time intervals between each shots will be calculated by the duration of the video and the number of total shots. It could be tuned by **−−timestep**, **−−time−from**, **−−time−end**, **−−opt−ffr**, and **−−opt−lfr** options.

Note that there are two **tricky** modes in this option. If *column* is 0, **ezthumb** will save screenshots into separate files. The suffix of these files would be numbers like **_001.jpg, _002.jpg, ...** etc. The total number of screenshots is decided by *row* . If *row* is 0 too, **ezthumb** will save all key frames (I-frame) into separate files. So watch out, it might be hundreds.

−G, **−−gui**

Force **ezthumb** to start in the GUI mode. Normally **ezthumb** works in batch mode if there are file names specified in the command line. The **−G** option can force **ezthumb** into GUI mode. All file names in the command line will be loaded into the list view.

−−gui−progress

Launch a progress dialog in the command line mode if the **ezthumb** was compiled with GUI enabled. This option could be useful while integrating with a file manager.

−i, **−−list**

Display the general information of media files in the list form. This option won't generate any pictures.

−I, **−−info**

Display the media information. This option won't generate any pictures.

−m, **−−format** *picture_format*

Specify the output format. **ezthumb** support three formats: *jpg, gif* and *png.* The *jpg* could be followed by an identifier @ and a quality factor. For example, the default format is *jpg@85* . The *gif* format can also be followed by a factor. In that case, it means to generate an **GIF animation** file. The factor is the time interval in millisecond between each frames. For example, *gif@1000* means to generate the animated GIF file from the screenshots, and each frame of the GIF file will be displayed by 1 second.

−o, **−−outdir** *directory*

Specify a *directory* for storing the output thumbnails. Normally the thumbnails would be stored into the same path of the source video. With this option, however, they can be stored into any directory. It could be helpful when generating thumbnails from **read-only** devices like CDROM.

−p, **−−process** *method*

Explicitly specify a method to generate thumbnails. **ezthumb** uses a seek-and-decode strategy called *'skim'* in default. It will seek to the most possible place in the video file and start to decode the nearest frame. If the video file doesn't support seeking, **ezthumb** will automatically switch to *'scan'* mode. *'scan'* is a single-pass scan strategy. It will scan the whole video file to find and decode the nearest frames. *'scan'* mode generally take longer time than the *'skim'* mode. If enabled the **−−accurate** option, it takes more time to scan through the P-frame. The *'2pass'* mode is similar to the *'scan'* mode. It will scan the video file twice to decide the best position to decode. *'key'* is used to rip key frames. It is almost same to **'−g 0x0'** except it can accept the number of ripped key frames. For example, *key@5* means to take first 5 key frames from the video clip. *'safe'* mode is based on the key frame ripping strategy. It decodes every key frames and picks up the closest frames to compose the thumbnails.

The default setting can process most of files. However, in dodgy situation, user might need to specify, or try, one of these strategies.

−P, **−−profile** *profile_string*
Specify a string of profiles to define the size of grid and screen shots. In simple, it can be treated as the combination of **−g** and **−s** options. Please see the **PROFILES** section below. Note that the **−P** option would override the internal profiles and configure file profiles, and the **−g** and **−s** options would override the **−P** option.

−R, **−−recursive** *suffix_filter*
Generate thumbnails from files under specified directories and subdirectories recursively. If no directory was specified, it would start from the current directory. A *suffix_filter* string is required to help **ezthumb** to filter out the target files. The string is composed by the extension filenames separated by comma. For example, *avi,rm,rmvb,mkv,wmv* . If an empty string was specified, like *−R ”* , **ezthumb** would try to process every files.

−s, **−−ssize** *thumbnail_size*
Specify the width and height of each screen shot. It could be defined by pixel like *160x120* or by a zoom ratio like *50%* . In latter case the screen shot would be 50% of the video frame size in either direction, such as 160x120 from a 320x240 video.

The pixel size doesn’t need to follow the original video ratio so that it could be adjusted to any aspect ratio as you wish.

Note that this option would be overriden by the **−w** option. If the **−w** was not specified, the dimension of the canvas would be summed up by the size of the screen shot, the grid size and the gaps between screen shots. However, if the **−w** option was specified, the width of each shot is deducted from the width of canvas and gaps. The height of each shot is calculated by the aspect ratio of the video.

−t, **−−timestep** *millisecond*
Specify the time interval between each screen shots in millisecond. If this option was not specified, **ezthumb** would calculate the average interval by the duration of the video and the number of total shots.

−w, **−−width** *size_in_pixel*
Specify the width of the thumbnail canvas in pixels. This option will override the **−s** option and calculate the size of each shots automatically.

−x, **−−suffix** *string*
Specify the suffix of the output file name. For example, if a video file named *myvideo.avi* and this option is set to **_pic ,** the file name of the generated thumbnail would be *myvideo_pic.avi*.

−−accurate
Let **ezthumb** take screen shots at more accurate places. Normally **ezthumb** would take screen shots at the nearest key frames (I-Frame). It is quick but inaccurate. With this option, **ezthumb** would take shots at both I-Frame and P-Frame. It is slow but accurate in time stamps. Sometimes the picture might be blurred.

−−background *filename*
Load a background picture into the thumbnail canvas. It supports **JPG** and **PNG** format. The background picture is placed in the canvas center by default. To adjust this, see **−−pos−bg** option for the details.

−−decode−otf *on|off*

Turn on or off the decoding-on-the-fly mode. In the scan mode, see the **−p** option, **ezthumb** does not decode the video frame while scanning. This option, on the other hand, would let **ezthumb** decode every key frames it met. The unused frame will be discarded after decoding. This option is aimed at some video clips which rely on previous key frames to decode a proper frame. This option is usually turned on to boost the compatibility. However it seems safe to turn it off with new version FFMPEG

−−depth *levels*

Descend at most *levels* (a non-negative integer) levels of directories below the command line arguments. **−−depth** *0* means unlimited.

−−edge *value*

Define the thickness of the frame edge around each screenshots. The thickness is defined by pixel size. The default value is 0 which means these's no frame edge.

−−filter *filter_string*

Specify the string of file extension name as a target file filter. For example, avi,flv,mkv,mov,mp4,mpg etc. The filter string is quite useful in **−R** recursive mode.

−−gap−shots *size_of_gap*

Define the gap size between the tiled screen shots. The size can be defined by pixel size or by percentage of the width of the screen shot. For example, *−−gap−shots 4* means the gap is 4 pixels between each screen shots. *−−gap−shots 4%* means the gap is 4% of of the width of a single screen shot.

−−gap−margin *size_of_margin*

Define the margin size around the thumbnail canvas. The size can be defined by pixel size or by percentage of the width of the screen shot. For example, *−−gap−margin 4* means to keep 4 pixels blank margin. *−−gap−margin 4%* means the margin is 4% of the width of a single screen shot.

−−opt−info *on|off|position_code*

Turn on or turn off displaying the media information on the top of the thumbnail canvas. If a *position_code* was set, it would be regarded as turning on the displaying. The default is *lt.* Note that the media information can only be placed on the top of the canvas.
See **POSITION CODES** for the details.

−−opt−time *on|off|position_code*

Turn on or turn off displaying the timestamp inside each screen shots. If a *position_code* was set, it would be regarded as turning on the displaying. The default is *lt.* Set up the position of the timestamp inside the screenshots. The default setting is *rt.*
See **POSITION CODES** for the details.

−−opt−ffr *on|off*

Turn on or turn off taking screen shots from the first frame. The default is *off* because most videos start from a black screen.

−−opt−lfr *on|off*

Turn on or turn off taking screen shots to the last frame. The default is *off* because most videos end at a black screen.

−−override *on|off|copy*

Turn on or turn off overriding existed thumbnails. The third option is *copy* which would generate thumbnails named by serial numbers. For example, if the thumbnail file should be named *video_file_thumb.jpg*, the *copy* option would generate *video_file_thumb.1.jpg* if *video_file_thumb.jpg* existed, or *video_file_thumb.2.jpg* if *video_file_thumb.1.jpg* existed, etc. The default is *copy* option.

Note the serial number is limited to **255.** If the limitation is reached, the last one, which should be *video_file_thumb.255.jpg* for example, will be overriden.

−−pos−bg *position_code (: qualification)*

Set up the position of the background picture. The default setting is *mc*.
See **POSITION CODES** for the details.

−−size−unit *auto|byte|kb|mb|gb*

Set up the unit of the video size in the output information. The unit can be *byte*, *kb* in kilobyte, *mb* in megabyte, *gb* in gigabyte, or *auto* for the most natural reading format.

−−time−from *starting_time*

Specify a time stamp from where the **ezthumb** will start to take shoots. The default setting is from the head of the video. The time stamp can be defined explicitly by *HH:MM:SS* form, or by the percentage of the video length like *33%* etc. Note that the **−−opt−ffr** and **−−opt−lfr** options are still applicable with this option.

−−time−end *ending_time*

Specify a time stamp to where the **ezthumb** will stop taking shoots. The default setting is the end of the video. The time stamp can be defined explicitly by *HH:MM:SS* form, or by the percentage of the video length like *66%* etc. Note that the **−−opt−ffr** and **−−opt−lfr** options are still applicable with this option.

−−transparent

Require to generate the transparent background which could be useful for the webpages. Note that only **PNG** and **GIF** support the transparent background.

−−factory−reset

Remove the configure file or the registry contents to recover the default settings.

−−vindex *video_stream_index*

specify the video stream index number inside the container file. The default behaviour of **ezthumb** is taking screen shots from the first video stream it has met. This option could override it and take screen shots from any stream. The stream indexes can be found by **−i** or **−I** option.

POSITION CODES

Position codes are used to describe the object position in the target image. There are ten position codes:

lt set the object to the left top corner

lc set the object to the left center side

lb set the object to the left bottom corner

mt set the object to the middle top side

mc set the object to the middle center

mb set the object to the middle bottom side

rt set the object to the right top corner

rc set the object to the right center side

rb set the object to the right bottom side

tt tile the object

For the background picture, the position code can be followed by a qualification code:

st stretch to fit the whole canvas

ex enlarge to fit the width of the canvas. The picture keeps its orignal ratio.

ey enlarge to fit the height of the canvas. The picture keeps its orignal ratio.

sx stretch the width of the picture to fit the canvas but keep its height same.

sy stretch the height of the picture to fit the canvas but keep its width same.

ENVIRONMENT

Since 3.0.0 ezthumb started to support an environment variable **EZTHUMB** to store some command line options. For example, the exported options by

export EZTHUMB="-g 4x6 -m png"

will be passed to **ezthumb** as default options. However, these options could be overriden by explicit command line options.

PROFILES

The profile is used to set a group of rules about the geometry size of the thumbnails. It can generate different screenshot according to the attribute of the video clip. For example, it can generate a 4x4 screen shot array if the video lasts 30 minute, or generate a 6x8 array if the video lasts 120 minutes. It can also generate a 240x120 thumbnail if the video frame is 160x120, or generate a 320x240 thumbnail if the video frame is 1920x1024.

In general words, there are two types of profiles. One is used to define the screen shot array by the length of the video. Another one is used to define the size of the thumbnails by the frame size of the videos. These profile entries can be combined by a *':'* like *12m4x4:30m4x8:90m4x16:320w100%:640w240x180:1280w20%* .

The profile entry has a fixed format

WEIGHT + flag + A + 'x' + B + 'x' + C

The *WEIGHT* is used to define the length of the video, or width of the video frame. **ezthumb** uses it to define the action range. For examples with the profile above, there are four ranges to make screen shot array by the length of videos: 0 to 12 minutes, 13 to 30 minutes, 31 to 90 minute and above 90. There are four ranges video frames to make different size of thumbnails, 0 to 320, 321 to 640, 640 to 1280 and above 1280 pixels. The *B* and *C* are optional. The *flag* is used to define the action inside the range.

M/m generate an A by B screen shot array according to the vide length in minutes

S/s generate an A by B screen shot array according to the vide length in seconds

L/l take shots by a logarithmic formula. The formula is

. lg(C)(length + A) - B

. The *C* is the base and the *length* is the video length in minutes. The screen shot array must be decided by combining with *F* or *R* flags.

W/w zoom the video frame by the specified ratio. *160W150* means to zoom in 150% of the original video frame.

T/t set the size of the thumbnail to the specified size. The *B* is optional so **ezthumb** would deduce it by video frame ratio.

F/f specifies the fixed canvas size. The parameter *A* is the thumbnails in a row. For example the *100F4x1280* generates a 1280 pixel picture with 4 thumbnails in a row.

R/r The size of the thumbnail fits best the specified width. The parameter *A* is the thumbnails in a row. For example the *100R4x320* generates 4 thumbnails in a row in the picture, each thumbnail has the size close to 320 pixel.

For examples, the default profile is *8M4x2:9L10x100x1.027:100R4x320* , which means if the video clip lasts 8 minute or less, it generates a 4x2 array; if the video clips lasts longer than 8 minutes, the number of generated thumbnail would be computed by the logarithm formula *ln(1.027)(length+10)-100* and the final picture would be 4 thumbnails in a row, each width would close to 320 pixels in width.

EXAMPLES

ezthumb −g 4x8 −s 33% *.avi

Create the 4x8 thumbnails for all *.avi* files in the current directory. Each screen shots inside the thumbnails are 33% of the video frame in width and height.

ezthumb −i *.avi

Display the length, the width and the height of all *.avi* files in the current directory.

ezthumb −g 1x12 −s 160x120 −−opt−ffr on −−opt−lfr on myvideo.avi

Create a 1x12 thumbnail where each screen shot is 160x120 pixels. Take the screen shots from the first frame to the last frame.

ezthumb −g 3x6 −w 1024 −t 60000 −−opt−info off −m png −−transparent myvideo.avi

Create a thumbnail in width of 1024 pixels with 3x6 screen shots inside. The size of each shots was calculated from this parameter. The interval between each shots is 60 seconds so it only took shots from first 18 minutes. Turn off the media information. The thumbnail is outputed in PNG format with a transparent background.

ezthumb −g 0x18 −s 120% −−opt−time off myvideo.avi

Generate 18 screen shots which were saved into 18 separated files. Each shots were 120% of the video frame in width and height. The inset timestamps were disabled.

ezthumb −g 3x6 −s 160x120 −m gif@1500 myvideo.avi

Generate an animated GIF file which include 18 frames. Each frame would be displayed by 1.5 seconds. The size of frames is 160x120. There is no canvas generated so the "-g" option is used for calculating the total shots only.

ezthumb −−accurate myvideo.avi

Generate a 4x4 thumbnail (the default "-g" parameter). Each shot is 50% of the video frame in width and height (the default "-s" parameter). The shots were taken in accurate mode so they could be taken as close as possible to the specified place.

AUTHOR

"Andy Xuming" <xuming@users.sourceforge.net>