Michael Pope
michael@dtcorp.com.au

Post Motion Detection

The problem

The cameras I've built using motioneyeos have some limitations

- Low frames per second
- Motion detection is done on the PI which is slow
- Motion detection cannot be altered

Original VLC solution

- Recording centrally using vlc

```
cvlc <camera url> --sout file/ts:#{path}.mjpeg
```

- Motion detection

```
xvfb-run -a cvlc --no-loop --play-and-exit --video-filter=motiondetect -vvv #{path}.mjpeg > #
h}.log 2>&1
```

Problems with original solution

- Slow
- Drops lots of frames and data
- Motion detection is limited and sketchy

New solution

- ENTER FFMPEG
- Centralise recordings from all cameras
- Do post motion detection on a decent server
- Capture at a higher resolution & frame rate.
- Text overlay on the server

The big scary commmand

```
ffmpeg -i http://10.1.1.175:8081 -vf "drawtext=fontfile=/usr/share/fonts/dejavu/DejaVu
Sans-Bold.ttf: \
text='%{localtime\:%T}': fontcolor=white@0.8: x=7: y=700: fontsize=24: box=1: boxcolor
=black" \
-vcodec libx264 -preset veryfast -f mp4 -pix_fmt yuv420p -y -f segment -segment_list o
ut.list \
-segment_time 3600 -segment_wrap 24 1-capture%03d.mp4
```

Break it down

-i	input
drawtext	Display the time in the bottom left, white on black
-vcodec libx264	Use x264 for the video output, good compression
-present veryfast	
-f mp4	container format
-pix$_{fmt}$ yuv420p	Select pixel format
-y	overwrite output files without asking
1-capture%03d.mp4	Name of the file with %03d being a 3digit number

Segment breakdown

-f segment	chop the video up
-segment$_{list}$	keep a list of segments (optional)
-segment$_{time}$	time in seconds for each file 3600 = 1hr
-segment$_{wrap}$	How many files to keep 24 * 3600 = 1day

Post Motion Detection

```
ffmpeg -i 1-capture000.mp4 -vf "select=gt(scene\,0.003),setpts=N/(25*TB)" 2-motion000.
mp4
```

Break down

-i	input
-vf	Create a filtergraph (allows the following)
gt(scenne\, %change)	% of changed pixels between 0 to 1 (0.003 is good)
setpts	change speed of video stream (0.5*PTS is double speed)

Demo

1. Run motion.sh with 0.003

```
~/ffmpeg_motion_test
./motion.sh
```

2. Run motion.sh with 0.006

Conclusion

This will allow electric eye to;

- do proper seamless recordings with post motion detection.
- Completely open source
- Easy to configure
- Keep track of multiple cameras

Questions

Email	map7777@gmail.com
Twitter	@map7
Github	github: map7