Subframe Video Post-Synchronization
1. Testpattern for Measuring Offset

Helmut Dersch
November 24, 2016

Abstract
This is a 3-part series of articles dealing with post-synchronization of video streams for stereoscopic applications, and/or for stitching panorama movies. The first two parts present methods for determining the offsets between video streams with sub-millisecond accuracy. The third part presents software tools for time-shifting videos by frame-interpolation establishing synchronized streams with the same accuracy.

Contents

1 Introduction 1
2 Testpattern 2
3 Testvideo 3
4 Results 4
5 Comparison with other methods 6

1 Introduction
Videocapture using several cameras like stereoscopic video or stitched videopanoramas requires synchronized streams: Frames from each camera have to be taken at exactly the same time otherwise stereoscopic vision suffers or stitching errors occur.

Aligning to integer frame numbers may result in delays of up to half of a frame duration (8.3ms for framerate 60fps) when using independent cameras. Often, this
results in visible errors. As an example, typical velocities of human movements (walking, gesturing, etc) are $1 - 2 \text{m/s} \approx 1 - 2 \text{cm}$ per halfframe, a car moving at $50 \text{km/h}$ is $12 \text{cm}$ ahead in the delayed image. These distances may be of the same order of magnitude as the left-right parallax responsible for stereovision.

Hardware synchronization solves this problem but is only available in some (quite expensive) cameras [1]. There are a number of ways to retrofit some cameras either with hardware sync, or the ability to at least start recording synchronously, which will not be reviewed here. We present a software solution which allows the user to synchronize unaligned video streams with drifting clocks. This first article deals with precisely measuring the time delay between several streams using a testpattern.

## 2 Testpattern

A simple technique for measuring time delays between video streams makes use of the special way cathode-ray-tubes (CRT) in monitors render images [4]. The position of the electron beam comprises a very accurate time stamp, and taking images of the moving beam using several cameras can be used to determine the time delay between cameras for this particular take.

CRTs are extinct now, and they can not easily be used outside in the field. A similar but less accurate method can be realized using fast OLED-displays on smartphones capable of displaying 60 frames per second (Samsung Note 4 in this work). These displays render images from top to bottom (portrait-mode) one row at a time. When the image is finished, the next frame is rendered immediately afterwards. The currently rendered line moves at a constant speed downwards and its position can be used as timestamp. An ideal test should consist of a serious of alternating black / white (or other high contrast) images making the currently rendered line easily identifiable as black/white border.

![Figure 1: Two consecutive frames of testvideo. The stream alternates between these two patterns.](image)
Figure 1 shows two consecutive frames of this pattern. In addition to the background color, a measuring grid and frame numbers are provided. 60 seconds of these frames alternating at a rate of 60 fps can be downloaded here [2]. The pattern is easily destroyed by modest compression levels, so we ask to not upload and distribute this video through Youtube or similar websites.

3 Testvideo

Video cameras usually have CMOS-imaging chips which exhibit rolling-shutters [3], i.e. the image is detected one row at a time (landscape mode for the cameras used in this work), similar to how the screen image for the test pattern is rendered. The apparent movement of the shutter overlaps the movement of the test pattern in the smartphone screen, and the final testvideo depends on the relative orientation of the two.

Figure 2: Frame of Testvideo. Rolling shutter moves perpendicular to screen image

Figure 2 shows results for the case when the two movements are perpendicular (rolling shutter moves from top to bottom, smartphone screen image from left to right). This leads to a slanted black-white border. The grid spacing horizontally represents 1/20 of a frame duration (0.83 ms for framerate 60 fps). The transition region extends about 3 ms horizontally which is due to the finite exposure time of this particular frame. The timing value can be read with better accuracy than this.
exposure duration, because there is one distinct point which is exposed equally by the white and black pattern, and which therefore is homogenously grey (zero-contrast).

In figure 3 results for parallel and antiparallel movement of shutter and rendered screen are shown. In these cases, the border is straight, and becomes widened or compressed. The perpendicular mode of figure 2 is simpler to analyze and has been used for the tests below.

It should be noted that the pattern observed in figures 2 and 3 slowly moves across the screen indicating that playback of 60fps videos on the Note 4 is actually somewhat slower, and occasionally, a frame is dropped. This introduces a tiny and negligible error in the timing values, as long as measurements do not coincide with frame dropping.

4 Results

The following results were obtained in a field test using two action cameras (Sony HDR AS100V/200V mounted as stereo-pair) and a Samsung Note 4 delivering the test pattern. Cameras are controlled using the Live-View Remote RM-LVR, and a recording session was started using the multi-camera feature of this setup. At the beginning and end of the recording interval the testpattern was briefly held into the view. Supposedly, the cameras should start recording simultaneously, but
frames with identical testpattern framenumber (2217 in the example below) are 12 frames (200ms) apart in the two recorded videos.

Figure 4 shows an extracted frame (Number 539) of the left camera, and the corresponding frame (Number 551) of the right camera. Analyzing the position of the slanted border between black and white regions enables us to determine the delay more precisely:

Figure 5 shows the border regions of both images enlarged. We have to measure time at the same rolling shutter position in both images, which means that the vertical pixel coordinate $y$ has to be the same in both images. Frame size is $1920 \times 1080$ pixels; I arbitrarily choose $y = 780$ which is indicated as horizontal yellow line in both images. We then have to find the intersection with the slanted border line, and count the horizontal units on the smartphone screen. I obtain about 2.6 units in the right and 2.3 units in the left image, i.e. a difference of 0.3 units. 1 unit corresponds to 1/10 of a frame duration (1.67ms), so the two images are 0.5ms apart. Since the frame numbers differ by 12, the total delay is $12.03 \times 16.7ms = 200.5ms$.

30 minutes, 108000 frames and 10km later this measurement is repeated and we get a delay of $11.25 frames \approx 187.5ms$. The internal clocks of the two cameras were drifting by $13ms/30min = 7.2 \times 10^{-6}$ apart. Given the rough environmental conditions (freezing temperatures, vibrations, bicycle mount) this is not unusual,
and shows that synchronized record start alone is not sufficient.

5 Comparison with other methods

The second article of this series presents a method for determining the same delay using the output of a software videostabilizer. This stabilizer calculates motion induced rotational angles for both cameras (yaw, pitch, roll). Since the two cameras are mounted rigidly to each other on a bicycle helmet, these movements are (almost) identical for both cameras, but time shifted by the delay. The delay then can be determined by a correlation analysis to subframe precision. Results are summarized in figure 6. The values determined by the testpattern method are shown as two circles at the beginning and end. The lines in between are determined by the correlation analysis of roll, yaw and pitch angles. The agreement is excellent.

Figure 6: Delay of left videostream with respect to the right stream as measured by the testpattern-method (circles), and videostabilizer (yaw-green, roll-cyan, pitch-blue). The video consists of 110000 frames at 60fps (30 minutes). Frame duration is 16.7ms.
Using a leastsquare linear fit we calculate the delay (in fractional frames) for each left-right framepair using the equation

\[ \Delta n = -7.2475 \times 10^{-6} \cdot n + 12.038 \]

This function is shown as red line in figure 6. Multiplying \( \Delta n \) by the frame duration \( \frac{1}{60} \text{s} = 16.7 \text{ms} \) yields the time delay with accuracy of approximately \( \pm 0.5 \text{ms} \).

References

[2] Homepage H. Dersch
[4] Camera Sync Tester software