Behaviour of realtime video mixers

W. Pasman, 31/1/1

Introduction

I planned to test the realtime behaviour of my software by mixing the graphics output in realtime with another video source. To do this properly, the latency of the video mixing devices should also be taken into account.
For this report I attempted to measure the latency of our digital video mixers (a Roland V-5 video canvas and a Vine Gen Pro). Unfortunately the efford quickly took another direction: what do these mixers do with my signals anyway? It showed that the Roland mixer does not keep the keep the order of the incoming data intact: sometimes frame parts come out in backward order. The Vine Gen drops parts of frames once in a while.

It would have been very helpful to know which chips are used in the converters. Unfortunately this information is not in the specification sheets, and a large number of manufacturers of such chips exist [AITechnologies01, Averlogic01, Dipix99, see also ePanorama00] which makes it difficult to guess it. Instead, I measured the performance of the converters.

In this report I discuss a few topics. I will first sketch the input sources I used and the way these were fed to the mixers. Next, I describe the way the outcoming mixed data was investigated. Then I discuss the measurement results, resulting in some ideas about the internal structure of these mixers and concrete latency values. I end with conclusions about their usability for realtime mixing.

Input sources

The video mixers I tested both need a **main input**. This main input is a standard PAL video signal (50Hz **fields** (halfframes of the interlaced video), roughly 720x576). This main signal is being mixed with another signal - the **mix input**, which can be of many formats, such as PAL or NTSC video, VGA XVGA or other computer generated formats and refresh rates. The mix input in this report was always VGA 640x480, 60 Hz, noninterlaced.
The output signal is always perfect in sync with the main input, thus is also of the 50Hz halfframes interlaced, 720x576 format. The sync signals of the mix inputs are not apparent at the output of the mixer (Figure 1). This is even the case if the mixer is set up to show only the mix input.

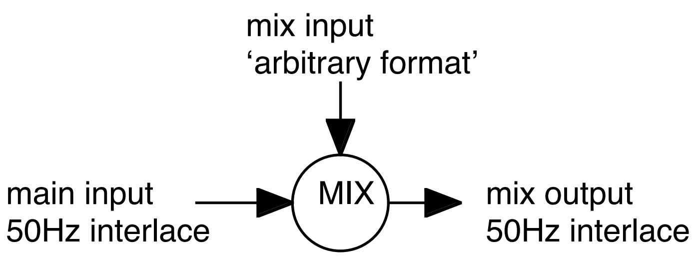


Figure 1. Mixer process diagram.

For this report I focused on the realtime behaviour of the mix input to mix output. I assumed - but did not test this explicitly - that the main input is never buffered but mixed in realtime with the mix input (Figure 2). In that case, the mix input has to be buffered because its raster position will generally be different from the main input.

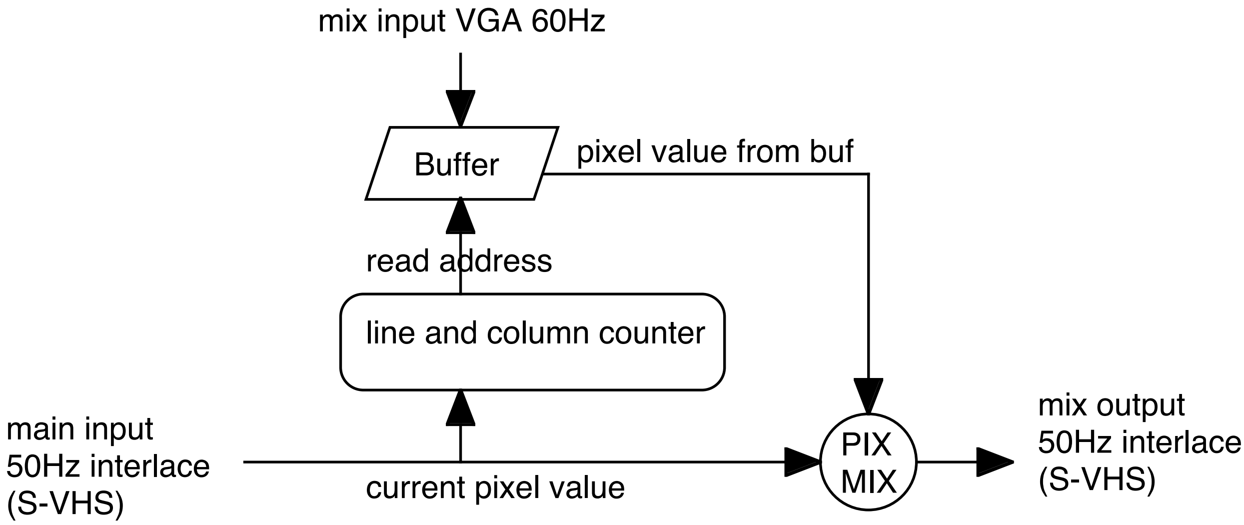


Figure 2. Assumed general mixer process diagram. The main input is never buffered but mixed on a per-pixel basis (PIX MIX) with the mix input. The mix input has to be buffered now.

I expected the buffer to be dual-port RAM, but I wanted to test this hypothesis. To test this, I generated a VGA sequence with 5 frames, to be played in a loop. As the VGA refresh rate is 60Hz, this loop is repeated 12 times per second. The VGA sequence consisted of 5 frames, each blank but of different color: red, yellow, green, cyan and dark blue in this order.

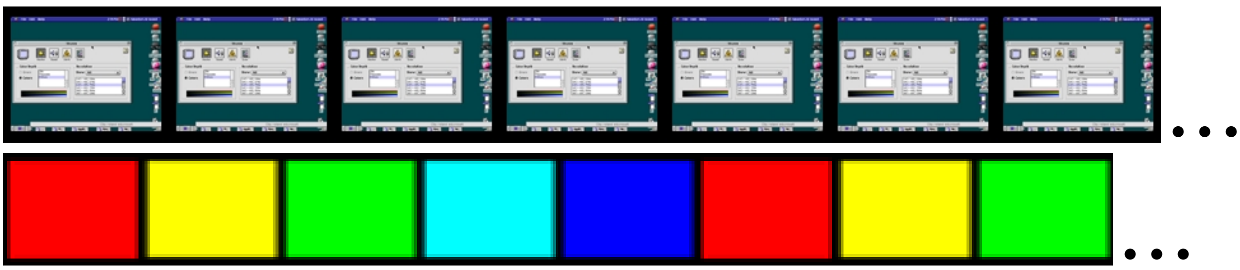


Figure 3. Two input signals. The top row shows seven **half-**frames at the main input (the output from the S-VHS output of my computer). The bottom row shows 8 VGA frames. The width of the frames is proportional to the time taken to display them. After 6 VGA frames the two are in sync again, but then with the next color as starting color.

If the buffer would be really a dual-port RAM, then a crack in the image would become visible approximately every 5th frame (Figure 4).

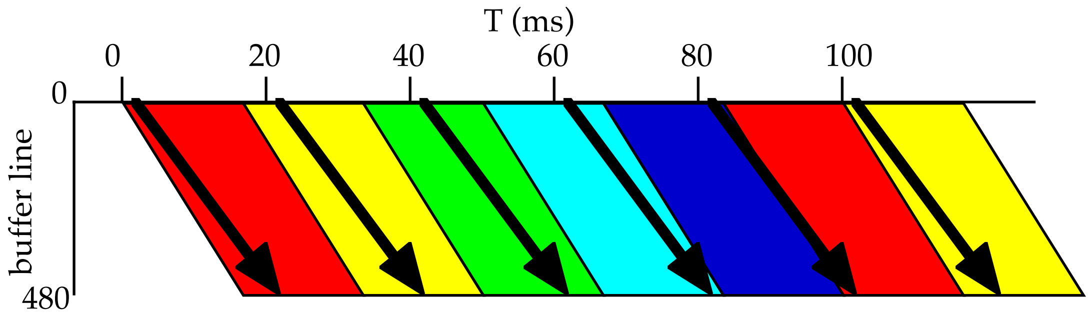


Figure 4. Black arrows indicate 50Hz readout of the buffer. Colored areas show buffer contents on a per-line basis. A crack in the image occurs where the black arrow crosses the border of two colored areas, here at approx. 90ms.

Investigation of mixed data

The mix output was sent to a JVC HR-DVS2 digital video deck and spooled to DV tape. In the first test setup, the DV tape was then spooled via firewire to a computer for detailed analysis. Adobe Premiere 5.1 was used to extract single frames from the video, and to split frames into halfframes. However, we found unexpected frames (Figure 5). It seems as if each frame contains 4 fields instead of normal two! On the video deck this was not visible. It seems that somewhere, maybe in Premiere, the DV-PAL 4:2:0 format (Figure 6, Jennings97) is incorrectly interpolated. Therefore we dismissed this analysis method. Instead, we viewed the frames directly on the video deck, which did not have this strange effect.

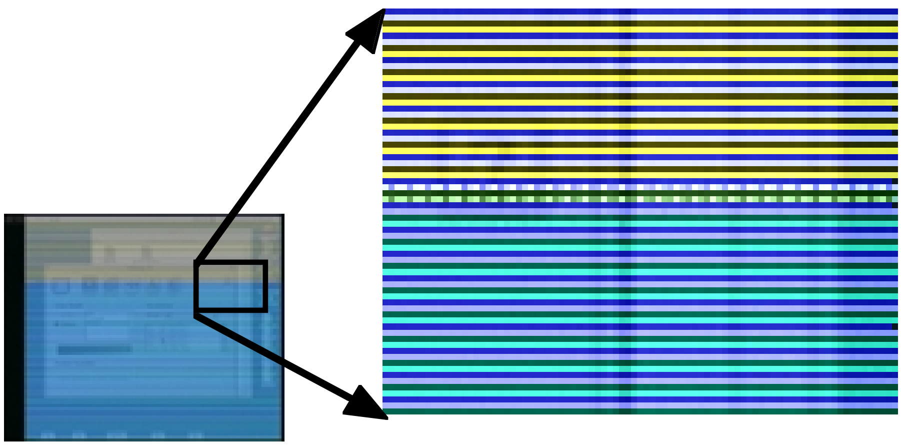


Figure 5. Example of the puzzling images captured from the DV tape to the computer using Premiere. Note that the pattern consists of 4 different colors instead of the expected two for an interlaced frame. The first and fourth line are OK, the other two seem strange.

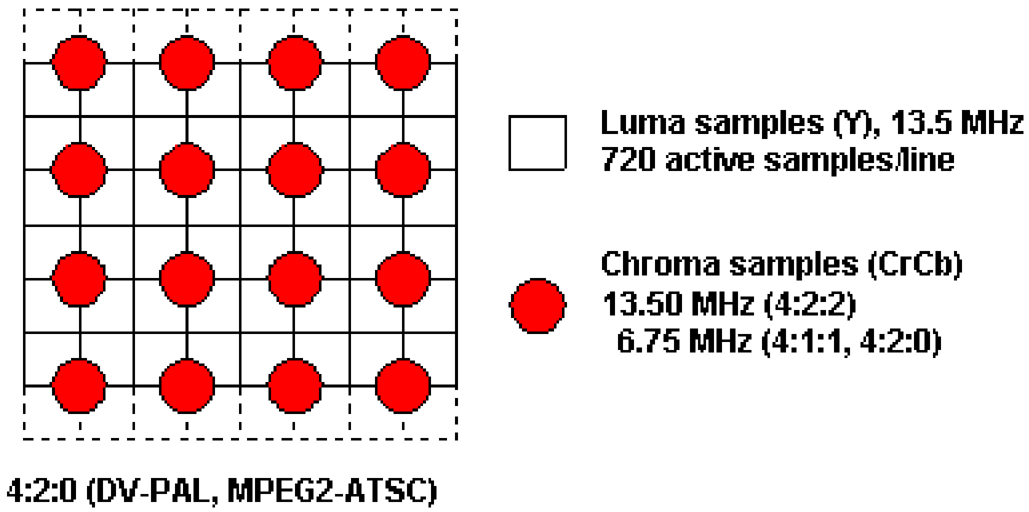


Figure 6. The 4:2:0 DV-PAL format has chroma samples only one of two lines, and just between the two lines as well. Interpolating may explain the strange effects.

Measuring mixer latency

In order to measure the latency of the mixers, I attached a 2-channel oscilloscope (Fluke PM3082 100MHz scope, 4 channels available). The VGA input was put on channel 1, and the S-VHS mixer output on channel 2. The VGA channel was modulated as follows: 1 field (halfframe) white, followed by 4 fields black. These fields can easily be recognised on the oscilloscope image. The scope was triggered on the upgoing flank of the VGA signal. Figure 7 shows a typical oscilloscope image.
The latency now can easily be found, by checking the distance between the flanks of channel 1 and 2. I want to know the maximum distance between the flanks.

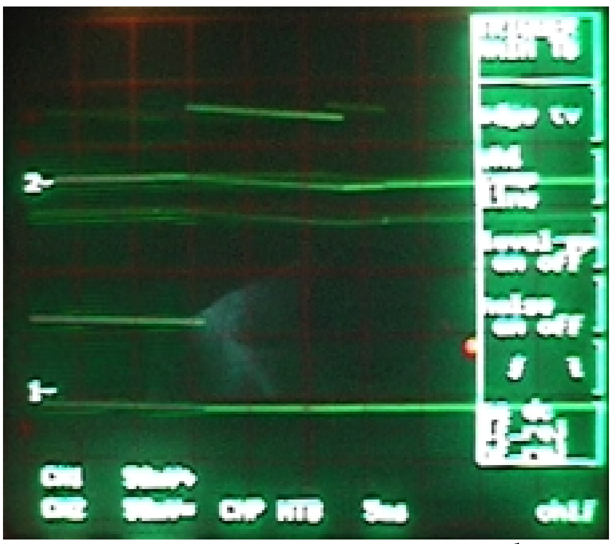


Figure 7. Typical oscilloscope image. Channel 1 (below) shows the VGA input, channel 2 the mixer output. (red dot is a reflection of a LED on the video camera)

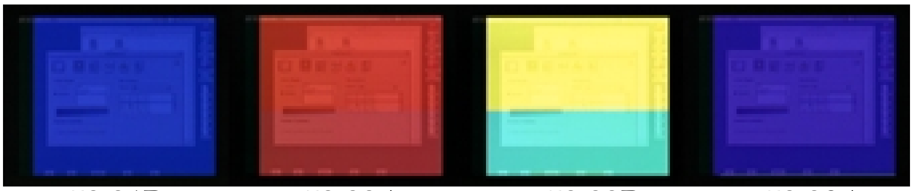
There are two problems with these images.
First, because the latency is highly fluctiating, the channel 2 image is not stable but all the time shifting from right to left or more random, depending on the mixer characteristics. This makes it particularly hard to estimate the maximum latency. To

get around this problem I recorded the scope image with a video camera, and investigated separate frames from video tape.
Second, the afterglow of the scope is quite long, long enough that several traces may be visible at the same time. In Figure 7 this can be already seen, as dim traces next to the relatively bright main trace. An advantage of this is that half-finished traces are fully visible in the next video frame.

Results for Roland V-5 mixer

For the Roland V-5 mixer we found some interesting results. First, sometimes a frame is dropped fully (Figure 8).

snapshot

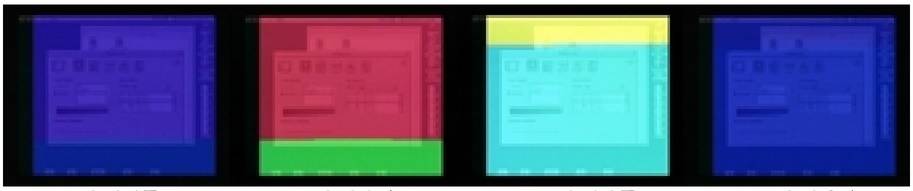


frame# 52.21B 52.22A 52.22B 52.23A

Figure 8. Snapshots from Roland V-5. Some frames are dropped entirely, here a green frame is dropped after the yellow frame.

Even stranger is that sometimes parts seem to play backwards (Figure 9). If there is a crack, the color after the crack sometimes skips one color, and the next frame may shows a short piece of the skipped color. So if there is a break we get a sequence ACBD where we expect ABCD. It shows that the longer fragment B, the shorter fragment C. It is very difficult to find the mechanism behind this. Instead, we proceed with the latency measurements first.

snapshot



frame# 52.21B 52.22A 52.22B 52.23A

Figure 9. Sometimes frameparts are shown in reverse order. Here we see a fragment of a green frame displayed before a yellow is displayed.

Latency measurements

We can already conclude something about the latency when looking at Figure 9. At the moment a bit of green appears in 52.22A, there must be a buffer containing the top of the yellow frame somewhere, otherwise this part could not show up in the next frame. Therefore the maximum latency is at least one full VGA frame.
A first artefact we encounter is the apparent presence of two signals on the mixer output: one of 16ms and another of 20ms (Figure 10). It seems that the 20ms is due to the camera, as this line is always travelling from left to right quickly. It always goes from left to right in 4 fields = 80ms, and travels 'through' the other signal. The higher signal, lasting roughly 18 ms, seems the real mixer output. The two remaining milliseconds are apparently lost in the vertical sync and borders.

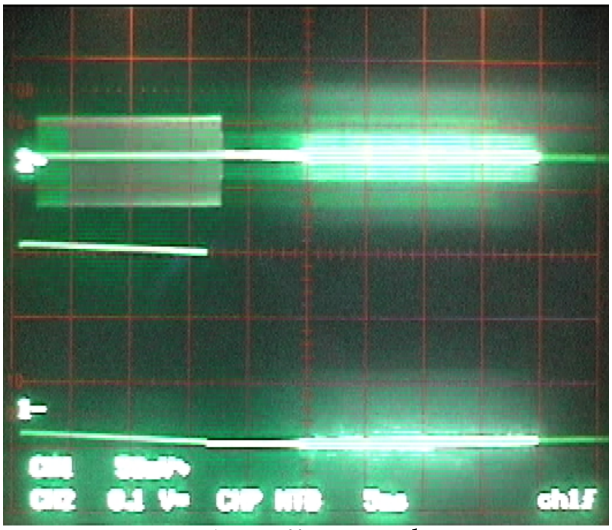


Figure 10. Peculiar effect: we have one signal lasting 20ms and another of 18ms on the mixer output. The 20ms "signal" seems due to the shutter speed of the camera, which was probably 20ms.

The mixer output shifts from left to right through the scope image. It seems to jump 16 ms to the right each time.

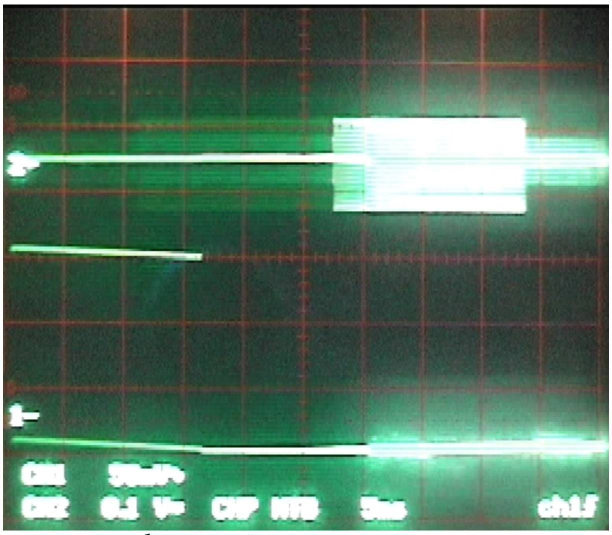


Figure 11. The signal is truncated at 43ms.

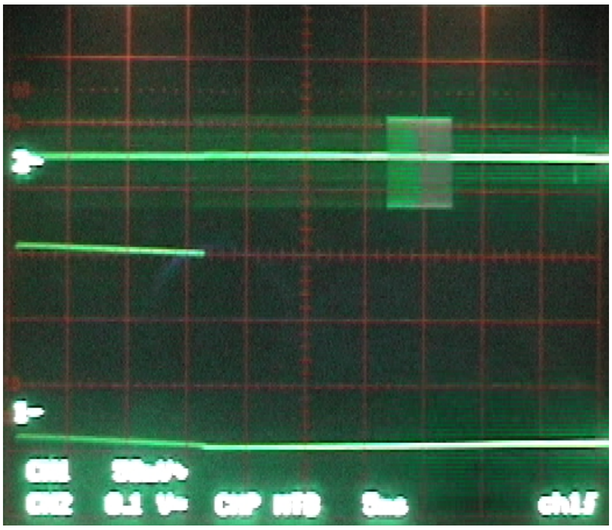


Figure 12. Here the signal is truncated already at 38ms.

To find the maximum latency, it seems easiest to use the right side and compare it with the end of the VGA signal. This rightmost points must correspond to a bottom line of the VGA signal. The largest distance I found here is shown in Figure 11, and is about 43 ms. Sometimes the signal is truncated earlier (Figure 12). This may have various reasons. The minimum latency is strange. Consider Figure 10, it seems that the latency here is only 2 or 3 ms.

Internal structure and processes

Given the behaviour and latencies found, it seems that the buffering uses either two fields of 20ms, or two VGA frames of 16ms. The minimum latency of 0ms also indicates that VGA data is not buffered first. But with these assumptions it is not possible to reconstruct the effects found. After a lot of testing of alternative solutions using various synchronisation mechanisms and double-buffering mechanisms, I found that a VGA buffer must be involved. The configuration of Figure 13 can indeed result in such behaviour. The VGA mixer input is first sampled fully into a VGA buffer. Once complete, the VGA buffer is copied quickly to the next field buffer (alternating the left and right field, not considering the current field used for reading). This system matches the actual behaviour if the copying takes approximately 3 ms.

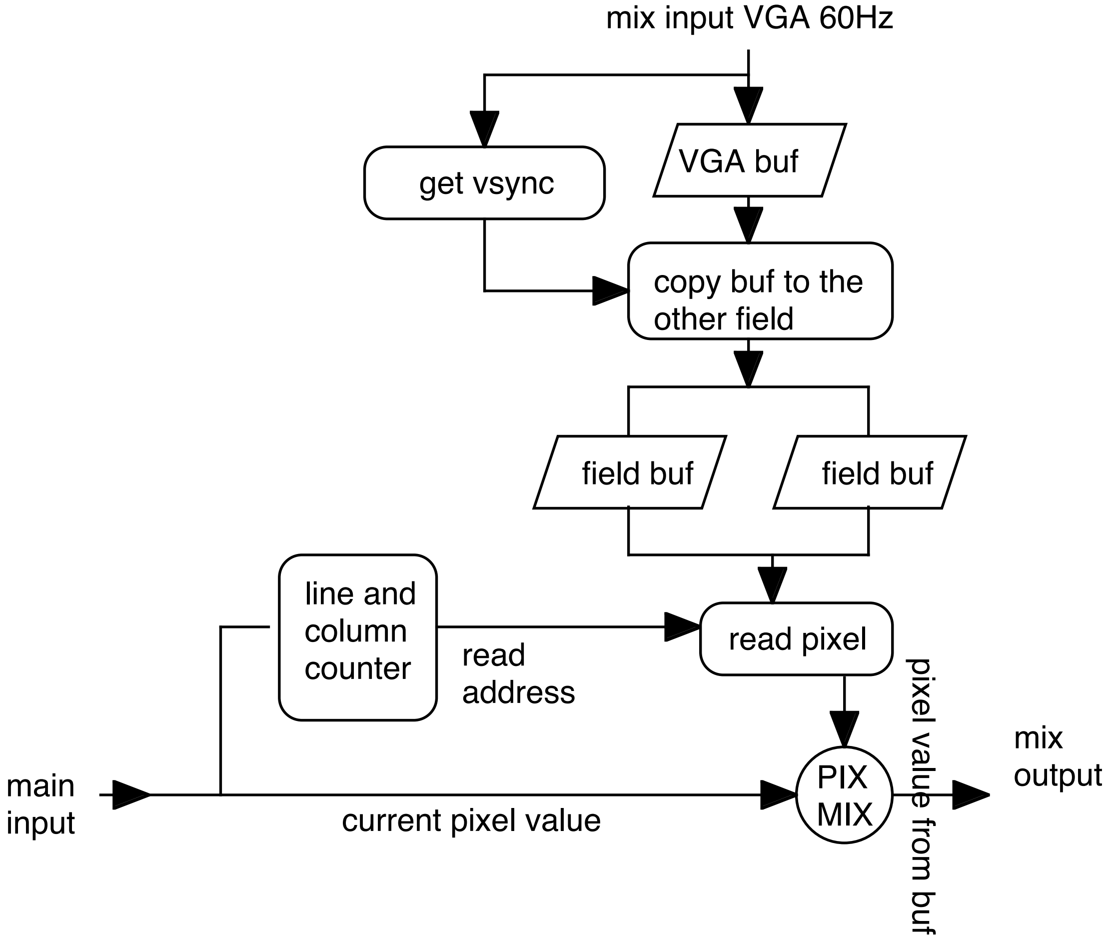


Figure 13. Reverse-engineered flow diagram of the Roland V-5 mixer.

Figure 14 shows that the spatio-temporal behaviour of this construct behaves as we found. The 'wrongly adjacent' parts in Figure 14 have to be about one-fifth of the 'correctly adjacent' parts. When the data is copied too slow, or sampled directly into the field buf, we do not get the reversed field parts or very small such parts. If it is copied faster, we would get much more cracks.

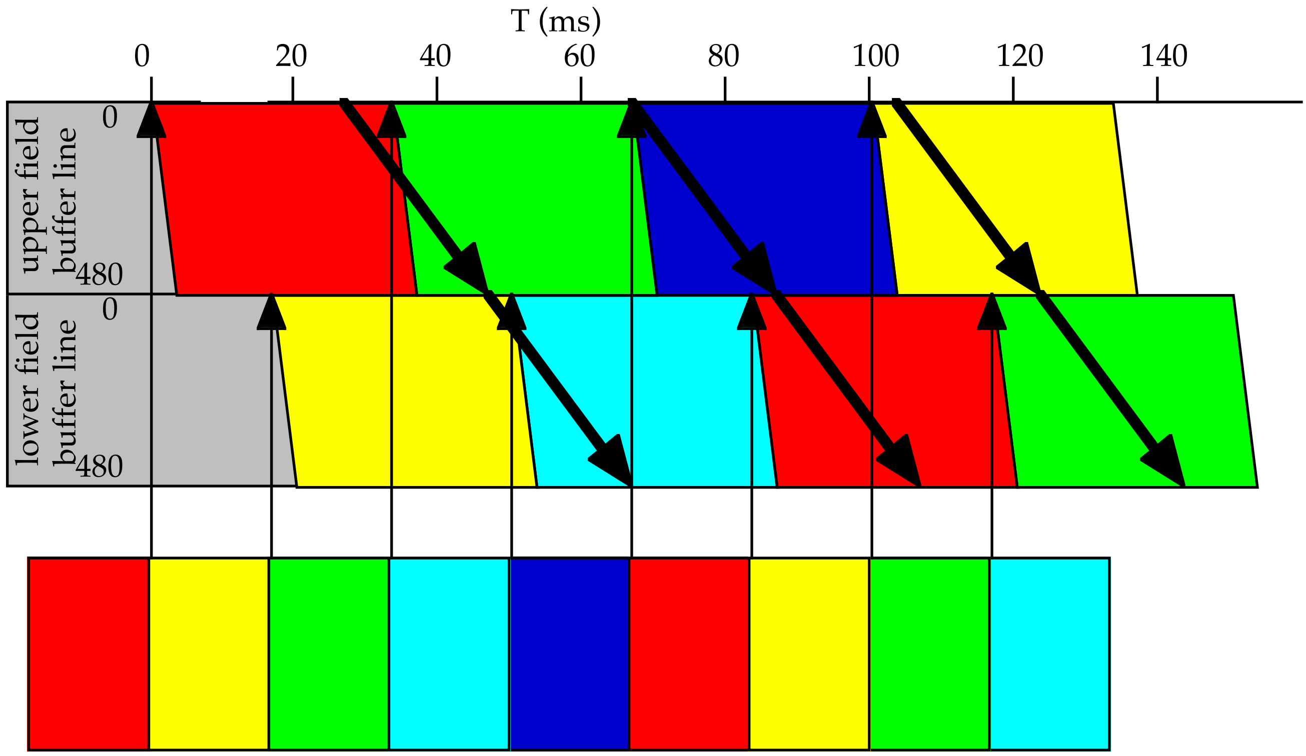


Figure 14. spatio-temporal behaviour of the Roland V-5 mixer, explaining Figure 9.

Note that this curious configuration requires twice as much dual-port RAM than our simple solution sketched in the introduction, and introduces twice as much artefacts as well!
One problem remains. Theoretically, the maximum latency following from this scheme is approximately 3 VGA frames = 3/60 s = 50ms, which is about 7ms more than the maximum of 43 ms we found in the measurements. The minimum latency is 3ms or the copy speed; this can be reached only for bottom lines and occur when the copy reaches the bottom line at the same time that the main input reaches the bottom line. But a counterexample is seen in Figure 10.

Results for Vine Gen Pro mixer

Again we have strange results: within a cycle of 5 frames, either one or two subsequent frames have a crack. Even stranger, often the first cracked frame in the cycle has only transparency below the crack (Figure 15), which replaces the color expected there. The second crack is often –if not always– higher in the field than the first crack.
If the crack is low enough in the field, part of the color becomes visible in the next field. Often, also the second cracked frame has transparency after the crack. The transparent cracks are always on 1/4 of the field, there are sometimes cracks on other heights but those are not so frequent and don't result in transparency below the crack.

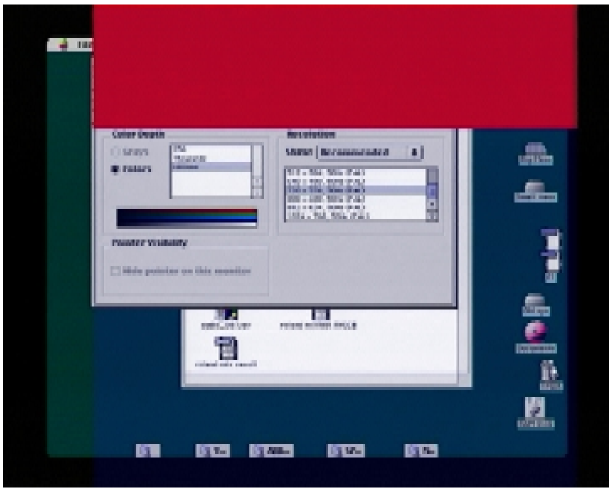


Figure 15. Transparency below a crack.

Another problem we have with this mixer is that we want the two inputs to be mixed transparently, that is added. This mixer can handle this, because it happens during so-called "fading" between input 1 and 2, but it seems not possible to stop half-way in the fading, instead fading continues always until only one of the inputs is visible. Another option would be to use the 'soft-keying' option: pixels that are much darker than the key level are keyed out completely, and pixels that are much brighter are keyed in completely. In-between pixels are mixed transparently, as we need. However, if we want to use this we will need to use non-saturated images, where the brightness is within the mixers quite vague 'much darker .. much brighter' range.

Latency measurements
In this case the white frame did not cover the full screen, therefore the white signal will last only 16 ms and not the full 20ms of the field.
Figure 16 shows that we best look for the end of the frame, because the frame may be displayed only partially but we cannot be sure just from a single scope trace, because our picture may be shot just too early for the trace to finish on the scope.

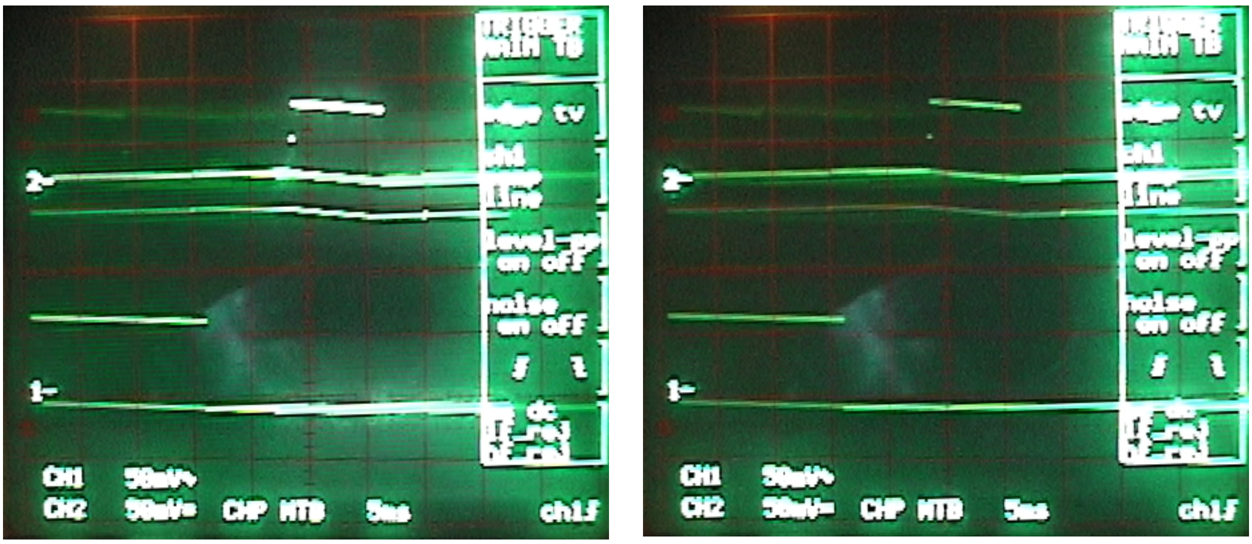


Figure 16. Left: some frames higher latencies than 20ms, but these are only partial frames, matching a half-black half-white screen (right).

At the high-latency end we find many of these partial-rendered traces that are really partial. In fact, high latency seems to imply that only partial image will be shown. As the latency becomes smaller in the subsequent frames, more of the field becomes visible (Figure 17).

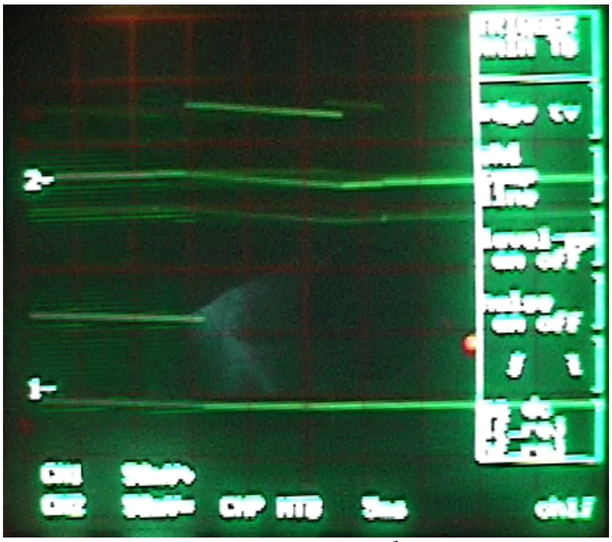


Figure 17. Slowly, more of the trace becomes visible, that is the channel 2 line shifts to the left over time.

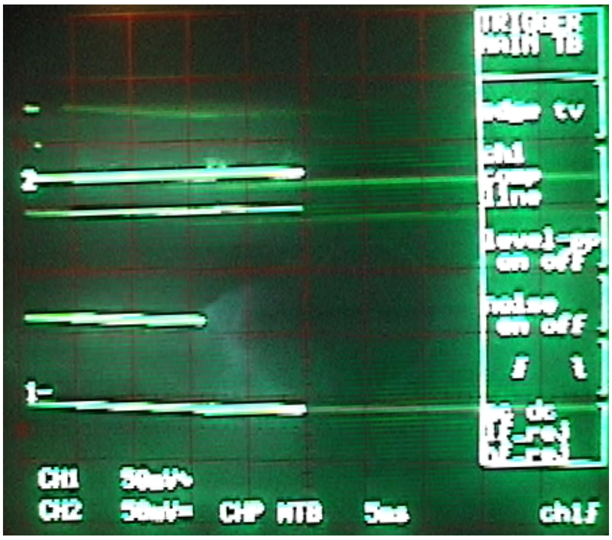


Figure 18. When it hits the left side (0ms latency), no full trace is achieved, just as with the right side. Apparently, the frame is cut out early now.

To find the maximum latency, it seems easiest to use the right side and compare it with the end of the VGA signal. The largest distance I found here is shown in Figure. The minimum latency behaves strange (Figure 18): it seems that only a partial trace is done then. Maybe this is related to the transparency-below-crack symptom of Figure 15.

As with the Roland mixer, we find rightmost point reached. The largest distance I found here is shown in Figure 19. The maximum latency found here thus is about 32 ms.

Figure 19. Maximum latency found.

I did not try to reverse-engineer this mixer as well. It took a lot of efford to find the Roland scheme, and the effects found here don't look really promising. Especially the dropped frame parts make this mixer a bad choice. Combined with its fixed location on the screen, this seems quite a low-level problem, maybe related to some internal memory size or erroneous forced synchronisation.

Other solutions

One option could be to implement a mixer in software, using for instance the SGI machines of the 11th floor. Jan-Roel Koppen and I tried to sample a VGA signal with this machine, but we could not get the A/D converters running properly at this rate.
A final possibility is to mix the signals optically, using two monitors and a half-transparent mirror. To prevent problems with the scan rates, LCD panels must be used. One way to do this would be to record with a video camera looking through our I-glasses protek helmet. A problem here is that we actually don't know the latency of the helmet. It is spec'ed for 60Hz VGA which suggests very low latency, but this was never checked.

Conclusions

If Roland and Vine Gen would have documented their mixers better, a lot of work could have been saved. After thorough measurement and reverse engineering, both mixers showed to have substantial problems. The Roland mixer displays frame parts sometimes in the wrong order, while the Vine Gen sometimes makes parts of frames invisible and can do the transparent mixing as we would like it without reducing the contrast range of the VGA images.
I never saw reversed frame parts with the Vine Gen, but a mis-timed image seems better than no image. This is one reason to prefer the Roland V-5 mixer over the Vine Gen for our mixing work. Second, Roland's errors in the temporal order of the frames will probably be acceptable, because they do not immediately result in visible reversed motions. This is because the temporal error is always between a fragment in the lower part and another fragment in the upper part of the fields.
A problem remains with correcting for the latency of the Roland mixer. A big problem point is the randomness of the latency: between 0 and 50 ms, but different for each frame.
Maybe optically mixing, for instance recording with a camera looking through our helmet, is an option. Theoretically this could result in very low latency, but the helmet latency was never estimated. Furthermore, such a setup makes our already quite complex setup even more complex. I planned to measure the performance of this option before making a final decision.

References

[AITechnologies01] AI Technologies, Inc. (2001). AIT2118 Description. AITech International Corporation, Fremont, CA. Available Internet: http://www.aitech.com/2108product.htm.
[Averlogic01] Averlogic (2001). AL128 data sheet. Averlogic Technologies, San Jose, CA. Available Internet: http://www.averlogic.com/data128.pdf.

[Dipix99] Dipix (1999). FPG-44 power grabber. Texas Instruments. Available Internet: http://www-d.connect.ti.com/dsp/tpcat/tpcodec.nsf/SoftwareForExternal/3004A050EE90FEAD8625670F005C7094.

[ePanorama99] ePanorama (2000?). Video chips and circuits. Available Internet: http://www.us-epanorama.net/videochips.html.

[Jennings97] Jennings, R. (1997). DV vs. Betacam SP: 4:1:1 vs. 4:2:2, Artefacts and other Controversies. In Special Edition Using Desktop Video, Que Books, Macmillan Computer Publishing. Available Internet: http://www.dvcentral.org/DV-Beta.html.