Measured latency of low-latency frontend

W.Pasman, 29/4/99

Introduction
In earlier reports it was discussed that augmented reality has severe constraints on the latency (Pasman, 1997). The end-to-end latency is the time between a movement of the observer and the corresponding update of the image in his helmet. Our first estimate was that 10ms would be an appropriate goal for the end-to-end latency.
This end-to-end latency can be split into several parts (Figure 1): the tracker latency, the rendering latency, the frame buffer latency and the display latency. To get a 10ms end-to-end latency, all parts in the chain of Figure 1 have to have such a small latency that their sum is 10ms. A low-latency rendering process is part of the solution.

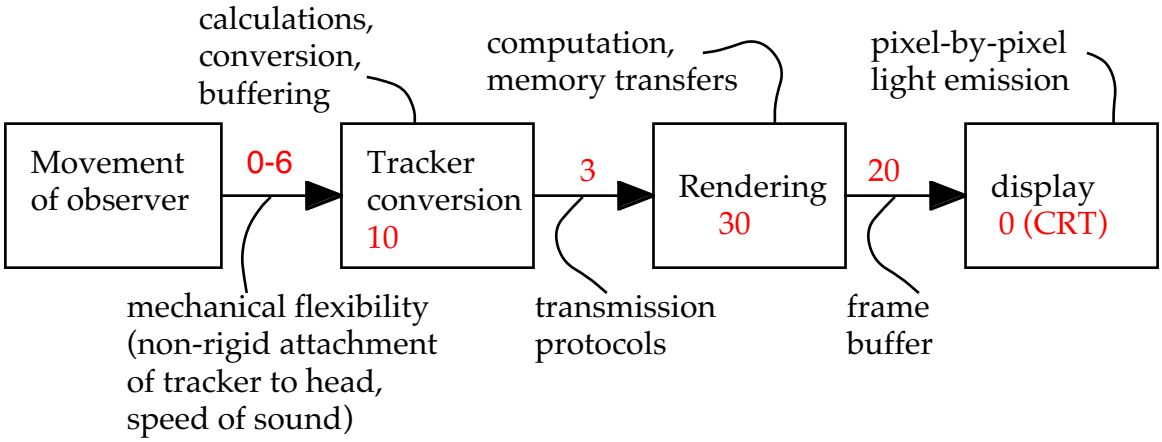

Figure 1. Sources of latency between observer movement and update of the display . Typical values for currently available, highly optimized systems are shown in red, in milliseconds.

A low-latency rendering system has been implemented by now (Pasman, 1999). It works by rendering only a quarter of the screen as close in time as possible before that part will be displayed. Theoretically it should attain a maximum latency of 1/120 sec, or 8.3ms. However, fluctuations in the scheduling, syncing accuracies and unforeseen delays in the hardware pipelines could increase the latency. This report measures the real latencies of the low-latency frontend.

What do we want to measure
The low-latency rendering system reduces both the rendering and frame buffer latencies. In the rest of this report I will refer to the total rendering and framebuffer latency as the **rendering latency**. As can be seen in Figure 1, the tracker measurements are the input parameter for the rendering. Thus, the rendering latency is the time between reading the latest position from the tracker and the output of pixels corresponding to that position on the vga output port of the graphics card.
Usually one tracker measurement is used to draw a large number of pixels. The minimum latency is the time from the moment the frontend acquires the latest tracker position/orientation

till the moment the first pixel appears on the vga port. The maximum latency is the time till the last pixel appears corresponding to that position.

Expected figures
Figure 2 shows the timings of the display output and rendering interrupts. To facilitate further discussion, we set t=0 at the end of the vsync pulse of the vga output. The vsync will trigger the rendering process, which should be finished by the time the display output arrives at the next visible area. A full frame including vsync is displayed in 1/60th second, or 16.67ms. If we partition this time in 4 we get rendering interrupts each 4.16ms. Each interrupt will trigger a render cycle. The display would run with the same rates if there were no vsyncs. However, the frame will be displayed in slightly less than 1/60th second, because it needs some additional time for the vsync, and so are the quarter frames. Finally, the rendering will start somewhere within the vsync, at T_r (which will have a negative value).

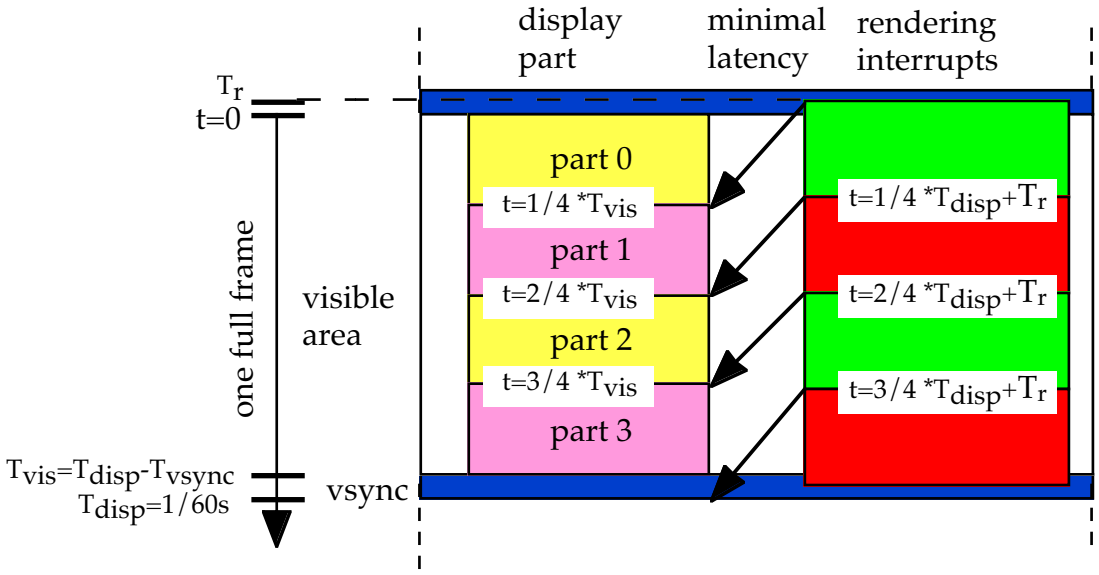

Figure 2. Expected timings of rendering and display. The T_{vsync} will be in the order of 1ms. see text.

The arrows in Figure 2 show the time between the moment the position is acquired till the moment the first pixel corresponding to that position becomes visible. This is called the minimal latency, because the other pixels in that part are displayed later and thus have a larger latency. The maximal latency is the time till the last pixel becomes visible, which is exactly the minimal latency + $T_{vis}/4$.
If rendering interrupts come regular, T_r is the only variable determining the latency. Table 1 sums up the minimal latencies for each of the display parts:

display part	minimal latency
1	$\frac{T_{vis}}{4} - T_r$
2	$\frac{2T_{vis} - T_{disp}}{4} - T_r$
3	$\frac{3T_{vis} - 2T_{disp}}{4} - T_r$
4=0	$\frac{4T_{vis} - 3T_{disp}}{4} + T_{vsync} - T_r = \frac{T_{disp}}{4} - T_r$

Table 1. Theoretical latency for each display part. Maximal latency=minimal latency+$T_{vis}/4$.

Technical setup

As discussed above, theoretically we only need to determine T_r. To measure it, I made the moment the frontend acquires the tracking position/orientation visible on the parallel port of the computer. As the vsyncs are already available on the VGA output pins of the computer, I could measure the time differences between those two pins with an oscilloscope.
To make the latency visible directly as well, a scene consisting of 1 white cube was created such that was not visible or filled the screen completely, depending on the orientation of the viewpoint. This way the screen would be either black or white depending on the orientation of the viewpoint, which is easy to find in the VGA signals.

An oscilloscope was used (Fluke PM3082, 100MHz bandwidth, 4 channel input) to trace these two signals. Both channels of the scope were triggered with a signal coming into channel 1 of the scope. I used only the first 2 channels of the scope. The channels measured the following:

Channel 1 was connected to the parallel port (pin 2=signal, pin 25=gnd). Pin 2 on the parallel port was steered by a special peace of software that replaced the standard positioning measurement system. Instead of reading tracker data, this stub did just 2 things:

```
1. n=n+1
2. if n=multiple of 4 then
      put 1 to parallel port pin 2
      return orientation such that object screen-filling
   else
      put 0 to parallel port pin 2
      return orientation such that object invisible
```

Channel 2 was connected to the VGA port of the graphics cards (pin 10=GND, pin 1=RED).

Results

Figure 3 shows the first result of the measurement. Before discussing the latency, I first have to explain the more basic things in the figure. Channel 1 shows the signal on the parallel port. The most interesting point for us is the moment when the orientation of the 'viewer' is such that the object should become visible, which is at the low-to-high transition of the signal. We can also see that the time channel 1 stays high, which is the time between two subsequent calls get the position, is $T_{disp}/4$ (one quarter of 1/60th second=4.16ms).
Channel 2 shows the signal on the RED pin of the VGA port. High indicates dark colors, low bright colors. The first dip of about 1.4ms is the vertical sync. Therefore $T_{vis}=T_{disp}-T_{vsync}$ =

15.3ms. The second dip, lasting $T_{vis}/4$ = 3.8ms, is the actual 'displaying' of the object (if we had a display connected instead of the scope).

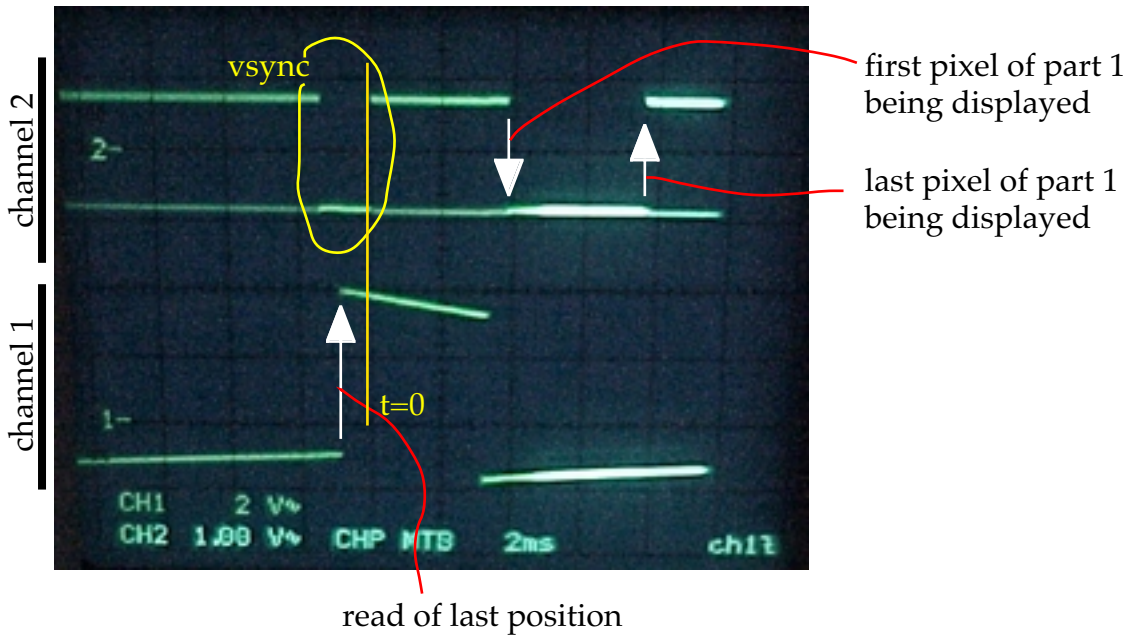

Figure 3. Scope screenshot. Channel 1 is the signal on the parallel port, channel 2 the red video signal. See text.

As can be seen in Figure 3, T_r ~ -0.8ms and T_{vsync} ~ 1.4ms. Filling in Table 1 gives minimal latencies for framepart 0(=4), 1, 2 and 3 of 5.0ms, 4.6ms, 4.3ms, and 3.9ms. Maximal latencies are then 8.8ms, 8.4, 8.1ms and 7.7ms. The average latency over the entire display is then 6.35ms, and the maximum latency is 8.8ms which occurs for the last pixel of the topmost framepart.

These latencies can also be seen directly in the scope pictures. Figure 4 shows the minimum and maximum latency for framepart 1 (framepart 0 is displayed directly after the vsync, and is black here). The time between the moment the position is known till the moment the first pixel corresponding to that position is displayed, and indeed shows to be about 4.6ms. The maximum latency is just 3.8ms larger, amounting to about 8.4ms as predicted.

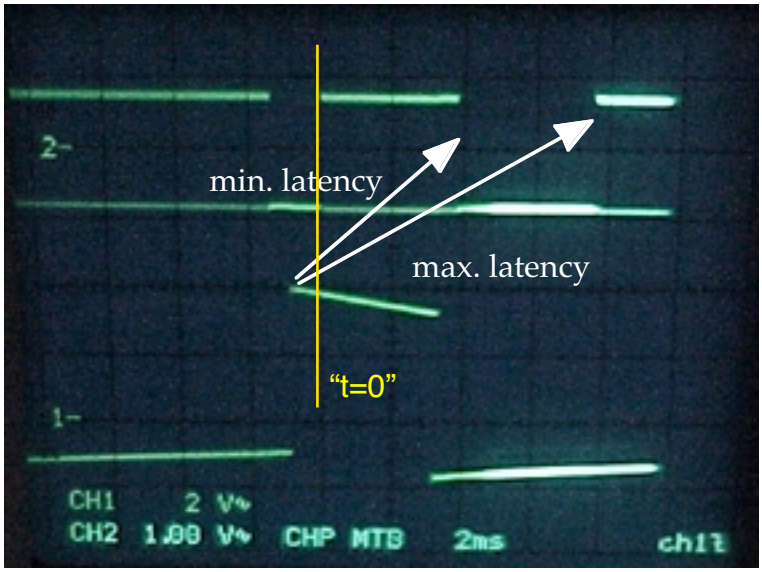

Figure 4. Minimum latency is 4.7ms, maximum latency 8.5ms.

Let's look at another framepart, framepart 3 (Figure 5). Here we should have the minimal latency. The vsync falls just at the end of the image, but we know where it should be because we already know its length (1.4ms). The minimal latency indeed is close to 3.9ms and the maximal latency close to 7.7ms.

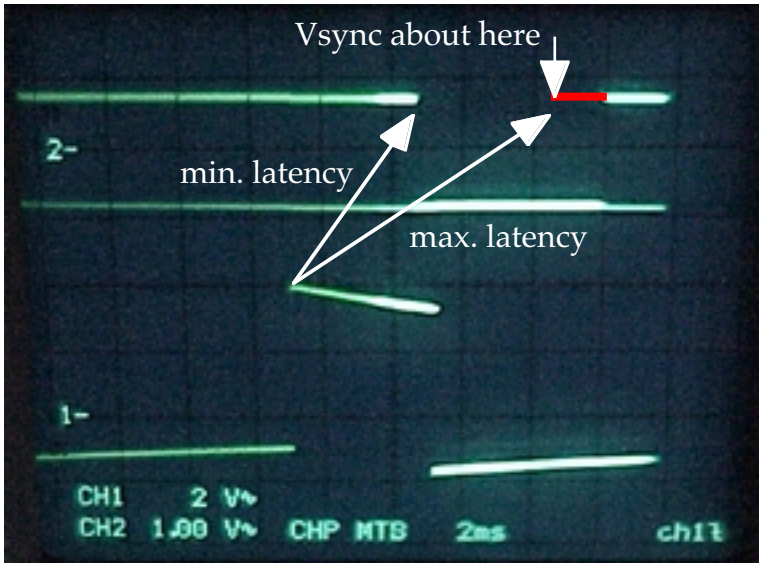

Figure 5. For framepart 3 the minimum latency is 3.9ms, maximum latency 7.7ms.

Conclusions

We showed that the theoretical latencies for the low-latency frontend have indeed be reached. The average latency over the entire display is 6.35ms, and the maximum latency is 8.8ms which occurs for the last pixel of the topmost framepart. With these values, an end-to-end latency of 10ms should be possible.

To get the end-to-end latency, the latencies of the other processes in the chain between head movement and display still have to be measured. Also, we will have to measure how many polygons can be rendered in the small time (less than 3.9ms) available for each framepart.

References

Pasman, W. (1997). Perceptual requirements and proposals for the UbiCom augmented reality display. Internal report, Delft University of Technology, Faculty of Information Systems and Technology, December. Available Internet: http://isis.et.tudelft.nl/bscw/bscw.cgi.

Pasman, W. (1999). Low-latency frontend implementation details. Internal report, Delft University of Technology, Faculty of Information Systems and Technology, May.