- Quick Start Guide
- Forum FAQ
- About
    - About us
    - Contact us
    - Technical, help and resource documents
    - Trademark rules

Search

- Login
- Register

-
    -
    - Unanswered topics
    - Active topics
    -
    - Quick Start Guide
    - Forum FAQ
    - About
        - About us
        - Contact us
        - Technical, help and resource documents
        - Trademark rules
- Board index Hardware and peripherals Camera board

ms latency between image acquisition and frame delivery

Post Reply

- Print view

Search this topic…

8 posts • Page **1** of **1**

aquilonis

Posts: 3
Joined: Sat Jan 07, 2017 7:14 am

ms latency between image acquisition and frame delivery

Quote

Sat Jan 07, 2017 7:40 am

Would anyone know the approximate latency between image acquisition and the delivery of image data to receiving software?

My reason for asking is that I'd like to sync a gyro/acc/mag data stream with camera images, so I can register sequential image frames (in a moving camera) without relying on image processing based registration. To do this accurately requires knowing how many milliseconds typically elapse between image acquisition and the delivery of the image frame to software.

For reference, I'm using an RPi3 board with a v2 camera, and I'm receiving YUV data through a RaspiStillYUV.c-based process.

As a side question, might anyone know if/when v2 cameras will be available that have their focal distances set to ~infinity (instead of ~1 meter)? I've purchased several v2 cameras and have to manually adjust the focal distance of each using needle-nose pliers, and I don't always do a good job...

Please accept my apologies if these questions has been asked/answered before.

jamesh

Raspberry Pi Engineer & Forum Moderator

ENGINEER

Posts: 26912
Joined: Sat Jul 30, 2011 7:41 pm

Re: ms latency between image acquisition and frame delivery

Quote

Sat Jan 07, 2017 11:38 am

It varies according to what you are doing. You can measure it. Get the Pi to display a MS timer, then record what its displaying using the camera, whilst displaying the preview in a small window. You will see the actual time and the time of the display - the difference is the delay.

Principal Software Engineer at Raspberry Pi (Trading) Ltd.
Contrary to popular belief, humorous signatures are allowed.
I've been saying "Mucho" to my Spanish friend a lot more lately. It means a lot to him.

6by9

Raspberry Pi Engineer & Forum Moderator
ENGINEER
Posts: 9377
Joined: Wed Dec 04, 2013 11:27 am
Location: ZZ9 Plural Z Alpha, aka just outside Cambridge.

Re: ms latency between image acquisition and frame delivery Quote

Sat Jan 07, 2017 9:57 pm

viewtopic.php?f=43&t=153410&p=1027417#p1004792 (https://www.raspberrypi.org/forums/viewtopic.php?f=43&t=153410&p=1027417#p1004792)

1280x960 YUV at 30fps, about 34ms from end of exposure of the first line to YUV frame delivered to app. Around 50ms for 1080P @ 30fps.

Software Engineer at Raspberry Pi Trading. Views expressed are still personal views.
I'm not interested in doing contracts for bespoke functionality - please don't ask.

aquilonis

Posts: 3
Joined: Sat Jan 07, 2017 7:14 am

Re: ms latency between image acquisition and frame delivery Quote

Sat Jan 07, 2017 11:14 pm

Thank you both very much!

On the topic of measuring latency (eg, when using other image configurations), but doing so in headless mode, does the following strategy sound reasonable?

1) Modify RaspiStillYUV.c
2) Using a high-resolution clock (eg, gettimeofday() or clock_gettime()), take the system time right before the image is requested from the camera*
3) Get system time again when camera_buffer_callback() function is called
4) Take the difference between the two times

* the request appears to be when "mmal_port_parameter_set_boolean(camera_still_port, MMAL_PARAMETER_CAPTURE, 1)" is called

Would you know the minimum delay between requesting the frame from the camera and when the start (end?) of exposure is, more or less, and how variable the delay is?

Variable delays can be compensated for by repeating the measurement and using the shortest interval, which should be the latency between exposure start (end?) and the YUV frame being delivered. Any constant delay would clearly bias the measurement, but hopefully such a delay would be very small...?

jamesh

Raspberry Pi Engineer & Forum Moderator

Posts: 26912
Joined: Sat Jul 30, 2011 7:41 pm

Re: ms latency between image acquisition and frame delivery Quote

Mon Jan 09, 2017 1:59 pm

There is quite a bit of work that goes on between asking for a capture and it happening. It needs to switch from preview mode to capture mode, then do the capture, switch back etc. That can add some variability. There are ways to get round this - that 6x9 may be able to help with as I cannot remember!

Principal Software Engineer at Raspberry Pi (Trading) Ltd.
Contrary to popular belief, humorous signatures are allowed.

I've been saying "Mucho" to my Spanish friend a lot more lately. It means a lot to him.

dmartel

Posts: 1
Joined: Fri Jan 13, 2017 6:18 pm

Re: ms latency between image acquisition and frame delivery Quote

Fri Jan 13, 2017 6:32 pm

I modified raspiyuv to capture what you are asking.

It seems it takes about 3x as the exposure rate is set to

6s exposure 18.61 seconds
1s exposure 3.26 seconds
0.5s exposure 1.76 seconds

The options I was using with the shutter speed (ss) changing changing are:

> Code:
>
> ```
> raspiyuv -y -v -o frame%08d.nv21 --width 640 --height 480 -ss 6000000 -n -ISO 100 -set -tl 0 -t 90000000 -ex off -awb off -ifx none -drc off
> ```

If the overhead was setting up the camera, switching modes, and writing to disk I would expect the overhead time to be constant. I am very curious to know why it is 3x the exposure time.

6by9

Raspberry Pi Engineer & Forum Moderator

ENGINEER

Posts: 9377
Joined: Wed Dec 04, 2013 11:27 am

Location: ZZ9 Plural Z Alpha, aka just outside Cambridge.

Re: ms latency between image acquisition and frame delivery Quote

Sat Jan 14, 2017 5:55 pm

It's been covered before, many times.

The sensor streams continuously regardless of when buffers are provided by the client.
On a capture request the current frame is completed, so up to one exposure time.
The sensor is then stopped and reprogrammed for the new 5MPix mode. Whenever the majority of rolling shutter sensors are restarted, the first frame is incorrectly exposed, so has to be discarded. Another exposure time.
You then need the sensor to exposure your desired frame. 3rd exposure time.

If the mode properties are the same for preview and the captures then the GPU code should avoid stopping the sensor, and therefore avoid having to discard the startup frame. If that isn't happening then I'll look into it.

Software Engineer at Raspberry Pi Trading. Views expressed are still personal views.
I'm not interested in doing contracts for bespoke functionality - please don't ask.

aquilonis

Posts: 3
Joined: Sat Jan 07, 2017 7:14 am

Re: ms latency between image acquisition and frame delivery Quote

Mon Jan 16, 2017 6:35 am

If we extend the question to 'what is the latency between frame request and frame acquisition start' then we can measure/estimate an answer to the original question, assuming that we know the shutter speed. So...

Per dmartel, time between frame request and delivery is approx 3x exposure time
Per 6by9, this is expected

Per jamesh, there is constant work that needs to be done between frame request and acquisition start (e.g., switching modes)
Per Murphy, there is randomness (i.e., Murphy's law)

Putting this together this implies that the latency between image request and the start of frame acquisition is approximately 2-times the frame exposure interval, time plus some "constant" interval (ie, constant + randomness).

If true, can one extend this to infer that the approximate latency between requesting an image and the start of the "capture" frame start time is, say, 2.1x the frame exposure time? Or is this simplifying the situation too much? E.g., if the camera determines its own exposure time then estimating latency based on frame exposure time is not very accurate?

Edit: The rephrasing of the question here is w/ the assumption that the camera is already powered up and is either streaming at X frames-per-second, or is waiting on a signal (eg, via SIGUSR1) to take acquire a frame.

Relatedly, is it possible to programmatically read out the exposure time for a given image frame?

Display posts from previous:
All posts

Sort by
Post time

Ascending

Go

Post Reply

- Print view

8 posts • Page **1** of **1**

Return to “Camera board”

Jump to

- Community
- General discussion
- Announcements
- Other languages
- Deutsch
- Español
- Français
- Italiano
- Nederlands
- 日本語
- Polski
- Português
- Русский
- Türkçe
- User groups and events
- The MagPi
- Using the Raspberry Pi
- Beginners
- Troubleshooting
- Advanced users
- Assistive technology and accessibility
- Education
- Picademy
- Teaching and learning resources
- Staffroom, classroom and projects
- Astro Pi
- Mathematica
- High Altitude Balloon
- Weather station
- Code Club
- Programming
- C/C++
- Java
- Python

  - Scratch
  - Other programming languages
    - Windows 10 for IoT
    - Wolfram Language
  - Bare metal, Assembly language
  - Graphics programming
    - OpenGLES
    - OpenVG
    - OpenMAX
  - General programming discussion
- Projects
  - Networking and servers
  - Automation, sensing and robotics
  - Graphics, sound and multimedia
  - Other projects
  - Media centres
    - Gaming
    - AIY Projects
- Hardware and peripherals
  - Camera board
  - Compute Module
  - Official Foundation Display
  - HATs and other add-ons
  - Device Tree
  - Interfacing (DSI, CSI, I2C, etc.)
- Operating system distributions
  - Raspberry Pi OS
  - Raspberry Pi Desktop for PC and Mac
  - Other
    - Android
    - Debian
    - FreeBSD
    - Gentoo
    - Linux Kernel
    - NetBSD
    - openSUSE

- Plan 9
- Puppy
- Arch
- Pidora / Fedora
- RISCOS
- Ubuntu
- Ye Olde Pi Shoppe
- For sale
- Wanted
- Off topic
- Off topic discussion

- Board index
- All times are UTC
- Delete cookies

ABOUT US

About us
Our team
Governance
Safeguarding
Our supporters
Jobs
Contact us

SUPPORT

Help
Documentation
Projects
Training
Downloads
Research
FAQ