



Arm Solutions at Lightspeed

Handling XR Displays in the Linux Graphics Stack

XDC 2026

Neil Armstrong - Jun Nie

Problem Description



What is an XR display

- Physically, an XR device consists of two distinct displays
 - one dedicated viewport for each eye
- To create stereoscopic depth, image must render a slightly offset perspective for both screens
- These panels demand exceptionally high refresh rates and ultra-low motion-to-photon latency to prevent user nausea
- Before final output, the XR compositor must apply complex optical distortion correction to account for the headset's physical lenses

Physical Characteristic



How is represented

- DRM doesn't define a display type
 - DRM doesn't really defines the monitor, only the connector capabilities
 - What's connected is a black box
- DRM supports “cloned” monitors
 - Considers the 2 display side-by-side
 - Still would be actual computer monitors
- DRM can't expose advanced monitor characteristics
 - Only physical width & height for embedded panels
 - Whatever is in the EDID or DisplayID blob

Why is it a problem ?

- It doesn't match anything right now
- The user-space DRM client can't know it's an XR display
- There's no standard way to expose the physical characteristics
 - No standard location
 - No standard format
 - No standard way to configure display engine features
 - Color correction
 - Chromatic aberration correction
 - ...

Solution ?

linaro linaro linaro

Kernel

- Add special DRM properties to XR displays
 - To prevent traditional clients from treating them as conventional outputs
 - To provide support for XR compositors
- Enable XR clients to query these properties
 - Identify XR outputs
 - Physical characteristics and capabilities
- Allow in-kernel support of complex setups
 - Dual-panel setups
 - Chromatic Abberation Correction
 - Writeback support for the whole XR display
 - ...

Panel driver

The general XR display would consist with:

- 2 separate LCD/OLED Displays, with the exact same properties
- Each one connected to a display link (DSI, eDP, ...)
- Both links must be clock synchronised

Current Solution is to

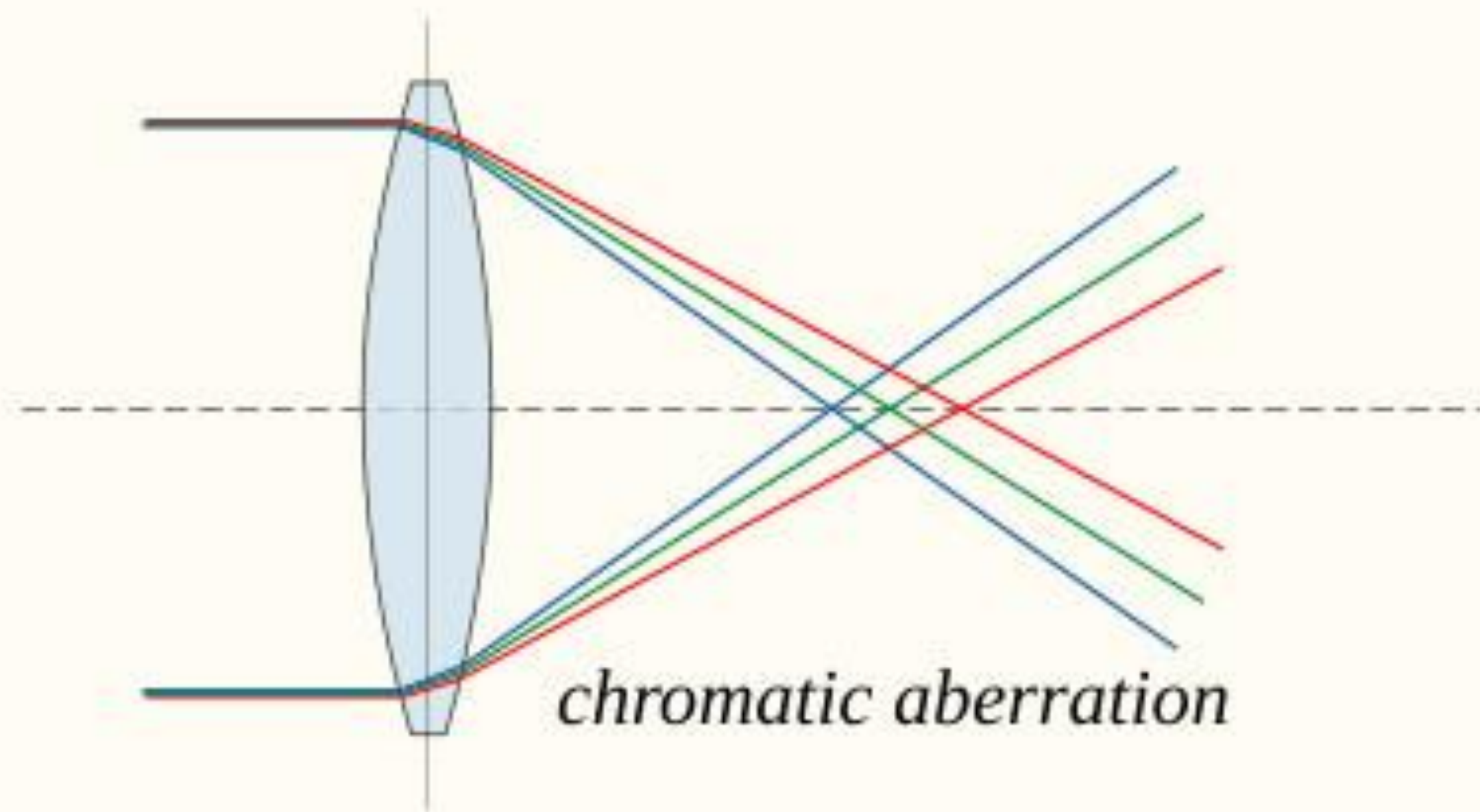
- Register a single panel for a dual-panel setup
- Mode is 2x the single panel mode
- Panel driver manages both panels
 - 2 set of DSI/power/backlight hidden inside the driver)
- Current proposal: <https://lkml.org/lkml/2026/7/27/548>

Chromatic aberration correction

Scales and shift pixels by R/G/B channel, based on lens and eye distance characters.

Implemented in either:

- display hardware as a feature
- display controller with planes & scaling
- in GPU composition



Expose CAC property

How to properly expose the correction tuning interface through DRM?

```
struct cac_property_blob {  
    /* source and dest rect of red/green/blue */  
    struct drm_rect dst_r, dst_g, dst_b;  
    struct drm_rect src_r, src_g, src_b;  
  
    /* rect for fetch from DDR memory */  
    struct drm_rect src_unpack;  
  
    /* CAC step for horizontal/vertical direction of R/B */  
    u32 phase_step_r_h, phase_step_r_v;  
    u32 phase_step_b_h, phase_step_b_v;  
  
    /* Initial CAC phase for horizontal/vertical direction of R/B */  
    u32 phase_init_r_h, phase_init_r_v;  
    u32 phase_init_b_h, phase_init_b_v;  
};
```

TODO

- Define what is a “Display” in DRM
- Allow defining an XR Display setup
 - Is the GPU the only hardware able to do the rendering transformations ?
 - Or should be consider those to be always done by User-space ?
- Allow exposing Physical characteristics
 - Via a well structured blob ?
 - Which set of values ?
- Allow programming the Chromatic aberration correction
 - Via a well structured blob ?
 - Which values ?



linaro.org

Thank you

Time for a short Q&A, we can discuss in the hallway for more questions !