Bug 1613260 - Support per-task scale for local space rasterization.

Bugzilla

Also: Bug 1613262 - Use local space rasterization for all off-screen surfaces

Contents

2020-02-14

Try to fix inflation and blur pushing the requestes size past `MAX_SURFACE_SIZE` again by calculating the scaling factor after we've taken all these into account. Which then means re-calculating the blur etc so move it into a helper function.

```
diff --git a/gfx/wr/webrender/src/picture.rs b/gfx/wr/webrender/src/picture.rs
index 480a1756504b..88b8d705021e 100644
--- a/gfx/wr/webrender/src/picture.rs
+++ b/gfx/wr/webrender/src/picture.rs
@@ -3776,11 +3776,11 @@ impl PicturePrimitive {
             Some(ref raster_config) => {
                 let pic_rect = PictureRect::from_untyped(&self.precise_local_rect.to_untyped());

-                let device_pixel_scale = frame_state
+                let mut device_pixel_scale = frame_state
                     .surfaces[raster_config.surface_index.0]
                     .device_pixel_scale;

-                let (clipped, unclipped) = match get_raster_rects(
+                let (mut clipped, mut unclipped) = match get_raster_rects(
```

```
16                      pic_rect,
17                      &map_pic_to_raster,
18                      &map_raster_to_world,
19  @@ -3794,11 +3794,59 @@ impl PicturePrimitive {
20                  };
21                  let transform = map_pic_to_raster.get_transform();
22
23  +               if raster_config.establishes_raster_root {
24  +                   println!("[BPE] establishes_raster_root");
25  +               }
26  +
27  +               fn adjust_scale_for_max_surface_size(
28  +                   raster_config: &RasterConfig,
29  +                   pic_rect: PictureRect,
30  +                   map_pic_to_raster: &SpaceMapper<PicturePixel, RasterPixel>,
31  +                   map_raster_to_world: &SpaceMapper<RasterPixel, WorldPixel>,
32  +                   clipped_prim_bounding_rect: WorldRect,
33  +                   device_pixel_scale : &mut DevicePixelScale,
34  +                   device_rect: &mut DeviceIntRect,
35  +                   unclipped: &mut DeviceRect) -> Option<f32>
36  +               {
37  +                   if raster_config.establishes_raster_root &&
38  +                       (device_rect.size.width  > (MAX_SURFACE_SIZE as i32) ||
39  +                       device_rect.size.height > (MAX_SURFACE_SIZE as i32))
40  +                   {
41  +                       // round_out will grow by 1 integer pixel if origin is on a
42  +                       // fractional position, so keep that margin for error with -1:
43  +                       let scale = (MAX_SURFACE_SIZE as f32 - 1.0) /
44  +                                   (i32::max(device_rect.size.width, device_rect.size.height) as f32);
45  +                       *device_pixel_scale = *device_pixel_scale * Scale::new(scale);
46  +                       let new_device_rect = device_rect.to_f32() * Scale::new(scale);
47  +                       println!("[BPE] scaling factor {:?}", scale);
48  +                       println!("[BPE] device_rect {:?}", new_device_rect);
49  +                       *device_rect = new_device_rect.round_out().try_cast::<i32>().unwrap();
50  +
51  +                       *unclipped = match get_raster_rects(
52  +                           pic_rect,
53  +                           &map_pic_to_raster,
54  +                           &map_raster_to_world,
55  +                           clipped_prim_bounding_rect,
56  +                           *device_pixel_scale
57  +                       ) {
58  +                           Some(info) => info.1,
59  +                           None => {
60  +                               return None
61  +                           }
62  +                       };
63  +                       Some(scale)
64  +                   }
65  +                   else
```

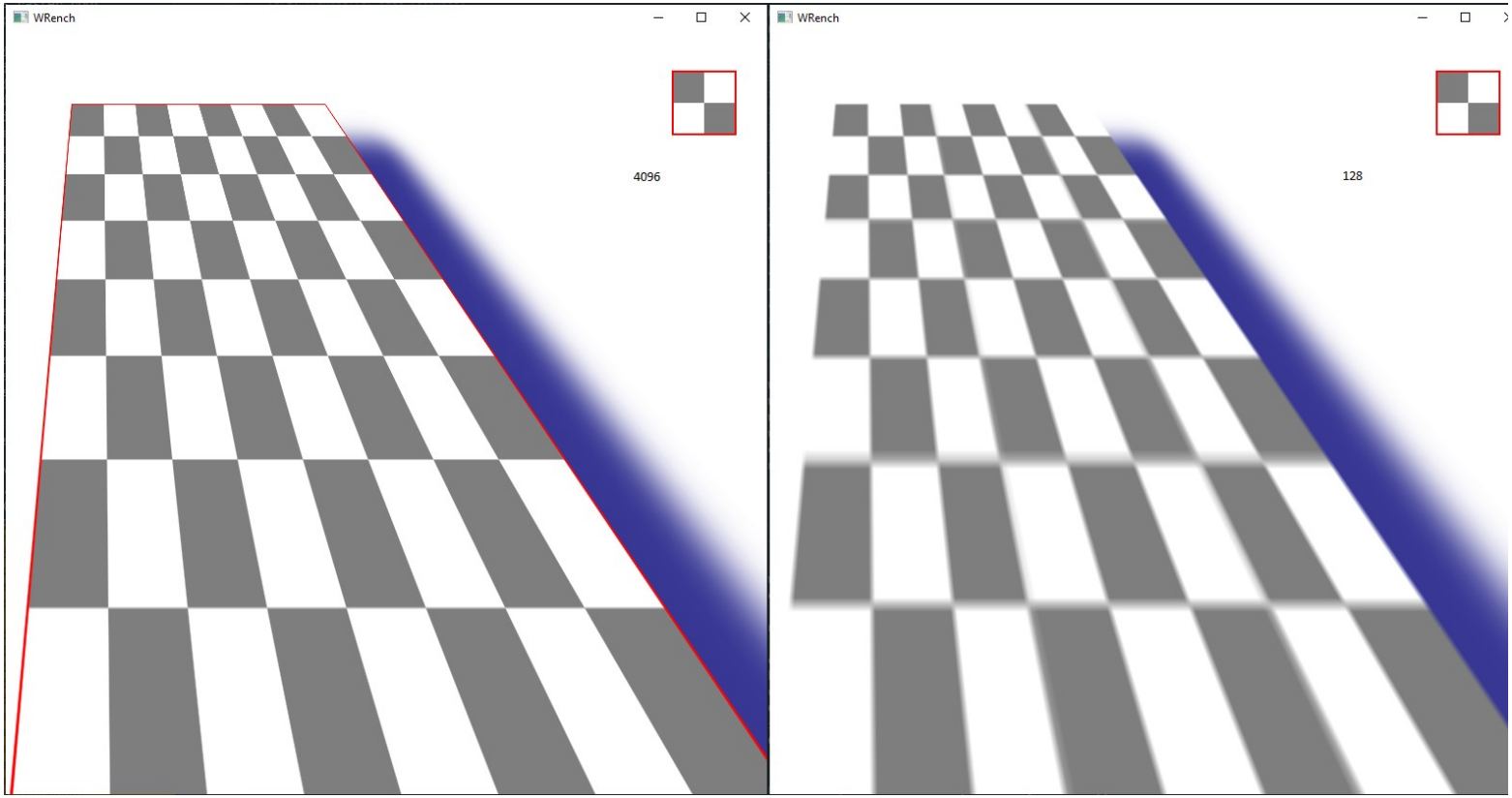
```
66 +                    {
67 +                        None
68 +                    }
69 +                }
70 +
71                  let dep_info = match raster_config.composite_mode {
72                      PictureCompositeMode::Filter(Filter::Blur(blur_radius)) => {
73                          let blur_std_deviation = blur_radius * device_pixel_scale.0;
74                          let scale_factors = scale_factors(&transform);
75 -                        let blur_std_deviation = DeviceSize::new(
76 +                        let mut blur_std_deviation = DeviceSize::new(
77                              blur_std_deviation * scale_factors.0,
78                              blur_std_deviation * scale_factors.1
79                          );
80 @@ -3831,7 +3879,7 @@ impl PicturePrimitive {
81                              clipped
82                          };
83
84 -                        let original_size = device_rect.size;
85 +                        let mut original_size = device_rect.size;
86
87                          // Adjust the size to avoid introducing sampling errors during the down-scaling passes.
88                          // what would be even better is to rasterize the picture at the down-scaled size
89 @@ -3841,6 +3889,15 @@ impl PicturePrimitive {
90                              blur_std_deviation,
91                          );
92
93 +                        if let Some(scale) = adjust_scale_for_max_surface_size(
94 +                                                &raster_config, pic_rect, &map_pic_to_raster, &map_raster_to_world,
95 +                                                clipped_prim_bounding_rect,
96 +                                                &mut device_pixel_scale, &mut device_rect, &mut unclipped)
97 +                        {
98 +                            blur_std_deviation = blur_std_deviation * scale;
99 +                            original_size = (original_size.to_f32() * scale).try_cast::<i32>().unwrap();
100 +                        }
101 +
102                          let uv_rect_kind = calculate_uv_rect_kind(
103                              &pic_rect,
104                              &transform,
105 @@ -3903,6 +3960,12 @@ impl PicturePrimitive {
106                              DeviceSize::new(max_std_deviation, max_std_deviation),
107                          );
108
109 +                        adjust_scale_for_max_surface_size(
110 +                            &raster_config, pic_rect, &map_pic_to_raster, &map_raster_to_world,
111 +                            clipped_prim_bounding_rect,
112 +                            &mut device_pixel_scale, &mut device_rect, &mut unclipped);
113 +                        // std_dev adjusts automatically from using device_pixel_scale
114 +
115                          let uv_rect_kind = calculate_uv_rect_kind(
```

```
116                             &pic_rect,
117                             &transform,
118 @@ -3986,6 +4049,12 @@ impl PicturePrimitive {
119                             Some((render_task_id, render_task_id))
120                         }
121                         PictureCompositeMode::Filter(..) => {
122 +
123 +                           adjust_scale_for_max_surface_size(
124 +                               &raster_config, pic_rect, &map_pic_to_raster, &map_raster_to_world,
125 +                               clipped_prim_bounding_rect,
126 +                               &mut device_pixel_scale, &mut clipped, &mut unclipped);
127 +
128                             let uv_rect_kind = calculate_uv_rect_kind(
129                                 &pic_rect,
130                                 &transform,
131 @@ -4010,6 +4079,11 @@ impl PicturePrimitive {
132                             Some((render_task_id, render_task_id))
133                         }
134                         PictureCompositeMode::ComponentTransferFilter(..) => {
135 +                           adjust_scale_for_max_surface_size(
136 +                               &raster_config, pic_rect, &map_pic_to_raster, &map_raster_to_world,
137 +                               clipped_prim_bounding_rect,
138 +                               &mut device_pixel_scale, &mut clipped, &mut unclipped);
139 +
140                             let uv_rect_kind = calculate_uv_rect_kind(
141                                 &pic_rect,
142                                 &transform,
143 @@ -4274,6 +4348,11 @@ impl PicturePrimitive {
144                         }
145                         PictureCompositeMode::MixBlend(..) |
146                         PictureCompositeMode::Blit(_) => {
147 +                           adjust_scale_for_max_surface_size(
148 +                               &raster_config, pic_rect, &map_pic_to_raster, &map_raster_to_world,
149 +                               clipped_prim_bounding_rect,
150 +                               &mut device_pixel_scale, &mut clipped, &mut unclipped);
151 +
152                             let uv_rect_kind = calculate_uv_rect_kind(
153                                 &pic_rect,
154                                 &transform,
155 @@ -4298,6 +4377,12 @@ impl PicturePrimitive {
156                             Some((render_task_id, render_task_id))
157                         }
158                         PictureCompositeMode::SvgFilter(ref primitives, ref filter_datas) => {
159 +
160 +                           adjust_scale_for_max_surface_size(
161 +                               &raster_config, pic_rect, &map_pic_to_raster, &map_raster_to_world,
162 +                               clipped_prim_bounding_rect,
163 +                               &mut device_pixel_scale, &mut clipped, &mut unclipped);
164 +
165                             let uv_rect_kind = calculate_uv_rect_kind(
```

```
166                                 &pic_rect,
167                                 &transform,
168 @@ -4858,14 +4943,6 @@ impl PicturePrimitive {
169                 let surface_index = state.pop_surface();
170                 debug_assert_eq!(surface_index, raster_config.surface_index);
171
172 -               // Check if any of the surfaces can't be rasterized in local space but want to.
173 -               if raster_config.establishes_raster_root
174 -                   && (surface_rect.size.width > MAX_SURFACE_SIZE
175 -                       || surface_rect.size.height > MAX_SURFACE_SIZE) {
176 -                   raster_config.establishes_raster_root = false;
177 -                   state.are_raster_roots_assigned = false;
178 -               }
179 -
180                 // Set the estimated and precise local rects. The precise local rect
181                 // may be changed again during frame visibility.
182                 self.estimated_local_rect = surface_rect;
```

Drop shadows

```
root:
  items:
    - type: "stacking-context"
      perspective: 100
      perspective-origin: 100 100
      items:
        - type: "stacking-context"
          filters: drop-shadow([73, 73], 20, [55, 55, 150, 1])
          transform: rotate-x(15)
          items:
            - image: checkerboard(2, 64, 8);
              bounds: [40, 40, 516, 516]
        - image: checkerboard(2, 32, 2);
          bounds: [700, 40, 68, 68]
```



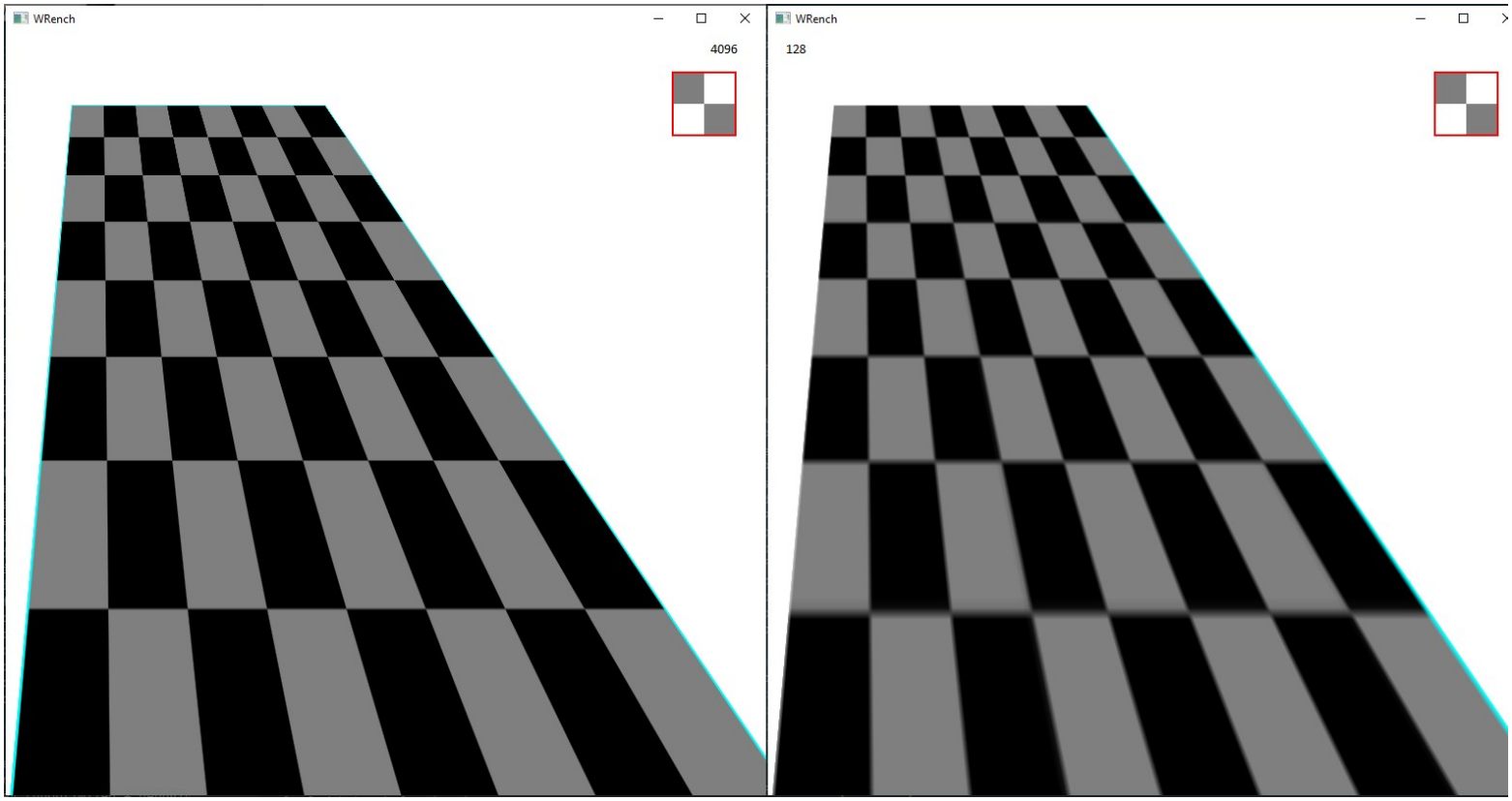

```
MAX_SURFACE_SIZE 4096

[BPE] TileCacheInstance::pre_update pic_rect Rect(2813.0×2358.0 at (-376.0,40.0))
[BPE] world_culling_rect Rect(1024.0×1024.0 at (0.0,0.0))
[BPE] PictureCompositeMode::TileCache::..
[BPE] establishes_raster_root
[BPE] PictureCompositeMode::Filter::DropShadows
[BPE] with_dynamic_location size 320×320
[BPE] with_dynamic_location size 160×160
[BPE] with_dynamic_location size 80×80
[BPE] with_dynamic_location size 80×80
[BPE] with_dynamic_location size 80×80
[BPE] pass 0 size (800x800)
[BPE] pass 1 size (800x800)
[BPE] pass 2 size (800x800)
[BPE] pass 3 size (800x800)
[BPE] pass 4 size (800x800)
[BPE] pass 5 size (800x800)
[BPE] pass 6 size (800x800)
[BPE] framebuffer size (800x800)
```

```
[BPE] pass 7 adds task location Fixed(Rect(800×800 at (0,0)))
[BPE] pass 6 adds task location PictureCache
        { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 0 }, size: 1024×512 }
[BPE] pass 6 adds task location PictureCache
        { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 1 }, size: 1024×512 }
[BPE] pass 0 adds task location Dynamic(None, 640×640)
[BPE] pass 1 adds task location Dynamic(None, 320×320)
[BPE] pass 2 adds task location Dynamic(None, 160×160)
[BPE] pass 3 adds task location Dynamic(None, 80×80)
[BPE] pass 4 adds task location Dynamic(None, 80×80)
[BPE] pass 5 adds task location Dynamic(None, 80×80)

MAX_SURFACE_SIZE 128

[BPE] TileCacheInstance::pre_update pic_rect Rect(2813.0×2358.0 at (-376.0,40.0))
[BPE] world_culling_rect Rect(1024.0×1024.0 at (0.0,0.0))
[BPE] PictureCompositeMode::TileCache::..
[BPE] establishes_raster_root
[BPE] PictureCompositeMode::Filter::DropShadows
[BPE] scaling factor 0.1984375
[BPE] device_rect Rect(127.0×127.0 at (-3.96875,-3.96875))
[BPE] with_dynamic_location size 128×128
[BPE] with_dynamic_location size 128×128
[BPE] pass 0 size (800x800)
[BPE] pass 1 size (800x800)
[BPE] pass 2 size (800x800)
[BPE] pass 3 size (800x800)
[BPE] framebuffer size (800x800)
[BPE] pass 4 adds task location Fixed(Rect(800×800 at (0,0)))
[BPE] pass 3 adds task location PictureCache
        { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 0 }, size: 1024×512 }
[BPE] pass 3 adds task location PictureCache
        { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 1 }, size: 1024×512 }
[BPE] pass 0 adds task location Dynamic(None, 128×128)
[BPE] pass 1 adds task location Dynamic(None, 128×128)
[BPE] pass 2 adds task location Dynamic(None, 128×128)
```

Invert filter

```yaml
root:
  items:
    - type: "stacking-context"
      perspective: 100
      perspective-origin: 100 100
      items:
        - type: "stacking-context"
          filters: invert(1)
          transform: rotate-x(15)
          items:
```

```
      - image: checkerboard(2, 64, 8);
        bounds: [40, 40, 516, 516]
    - image: checkerboard(2, 32, 2);
      bounds: [700, 40, 68, 68]
```



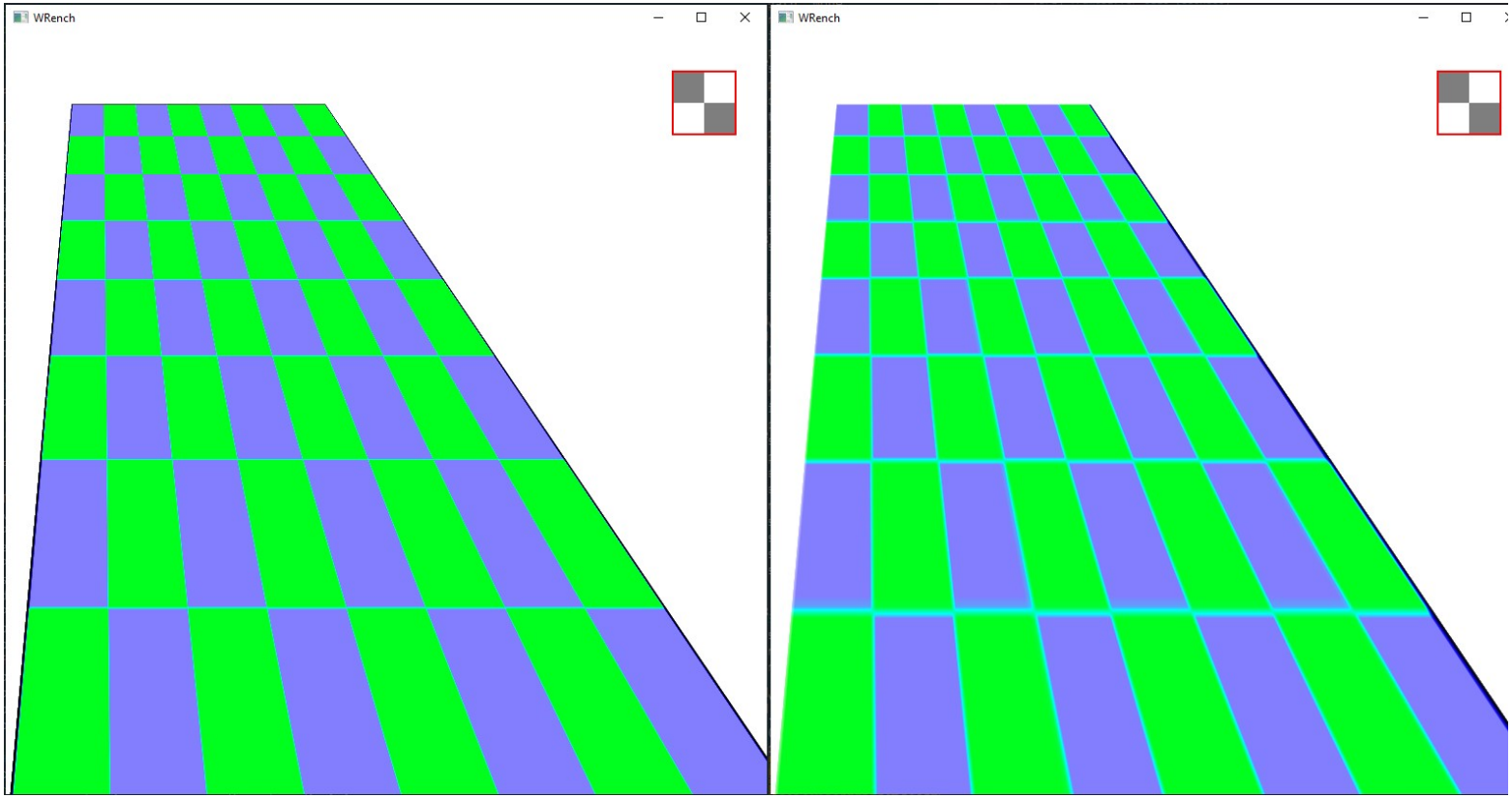

```
MAX_SURFACE_SIZE 4096

[BPE] TileCacheInstance::pre_update pic_rect Rect(866.0×816.0 at (-1.0,40.0))
[BPE] world_culling_rect Rect(1024.0×1024.0 at (0.0,0.0))
[BPE] PictureCompositeMode::TileCache::..
[BPE] establishes_raster_root
[BPE] PictureCompositeMode::Filter::..
[BPE] PictureCompositeMode::RTL:Dynamic due to Filter(..)
        clipped Rect(516×516 at (40,40)) unclipped Rect(516.0×516.0 at (40.0,40.0))
[BPE] pass 0 size (800x800)
[BPE] pass 1 size (800x800)
[BPE] framebuffer size (800x800)
[BPE] pass 2 adds task location Fixed(Rect(800×800 at (0,0)))
[BPE] pass 1 adds task location PictureCache
        { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 0 }, size: 1024×512 }
[BPE] pass 1 adds task location PictureCache
```

```
        { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 1 }, size: 1024×512 }
[BPE] pass 0 adds task location Dynamic(None, 516×516)

MAX_SURFACE_SIZE 128

[BPE] TileCacheInstance::pre_update pic_rect Rect(866.0×816.0 at (-1.0,40.0))
[BPE] world_culling_rect Rect(1024.0×1024.0 at (0.0,0.0))
[BPE] PictureCompositeMode::TileCache::..
[BPE] establishes_raster_root
[BPE] PictureCompositeMode::Filter::..
[BPE] scaling factor 0.24612403
[BPE] device_rect Rect(127.0×127.0 at (9.844961,9.844961))
[BPE] PictureCompositeMode::RTL:Dynamic due to Filter(..)
          clipped Rect(516×516 at (40,40)) unclipped Rect(128.0×128.0 at (9.0,9.0))
[BPE] pass 0 size (800x800)
[BPE] pass 1 size (800x800)
[BPE] framebuffer size (800x800)
[BPE] pass 2 adds task location Fixed(Rect(800×800 at (0,0)))
[BPE] pass 1 adds task location PictureCache
        { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 0 }, size: 1024×512 }
[BPE] pass 1 adds task location PictureCache
        { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 1 }, size: 1024×512 }
[BPE] pass 0 adds task location Dynamic(None, 128×128)
```

Component Transfer

```
root:
  items:
    - type: "stacking-context"
      perspective: 100
      perspective-origin: 100 100
      items:
        - type: "stacking-context"
          filters:
            - component-transfer
          filter-datas:
            - - - Table
                - Table
                - Table
                - Identity
              - - "1"
                - "1"
                - "0"
                - "0"
              - - "0"
                - "0"
                - "1"
                - "1"
              - - "0"
```

```
        - "1"
        - "1"
        - "0"
      - []
  transform: rotate-x(15)
  items:
    - image: checkerboard(2, 64, 8);
      bounds: [40, 40, 516, 516]
- image: checkerboard(2, 32, 2);
  bounds: [700, 40, 68, 68]
```



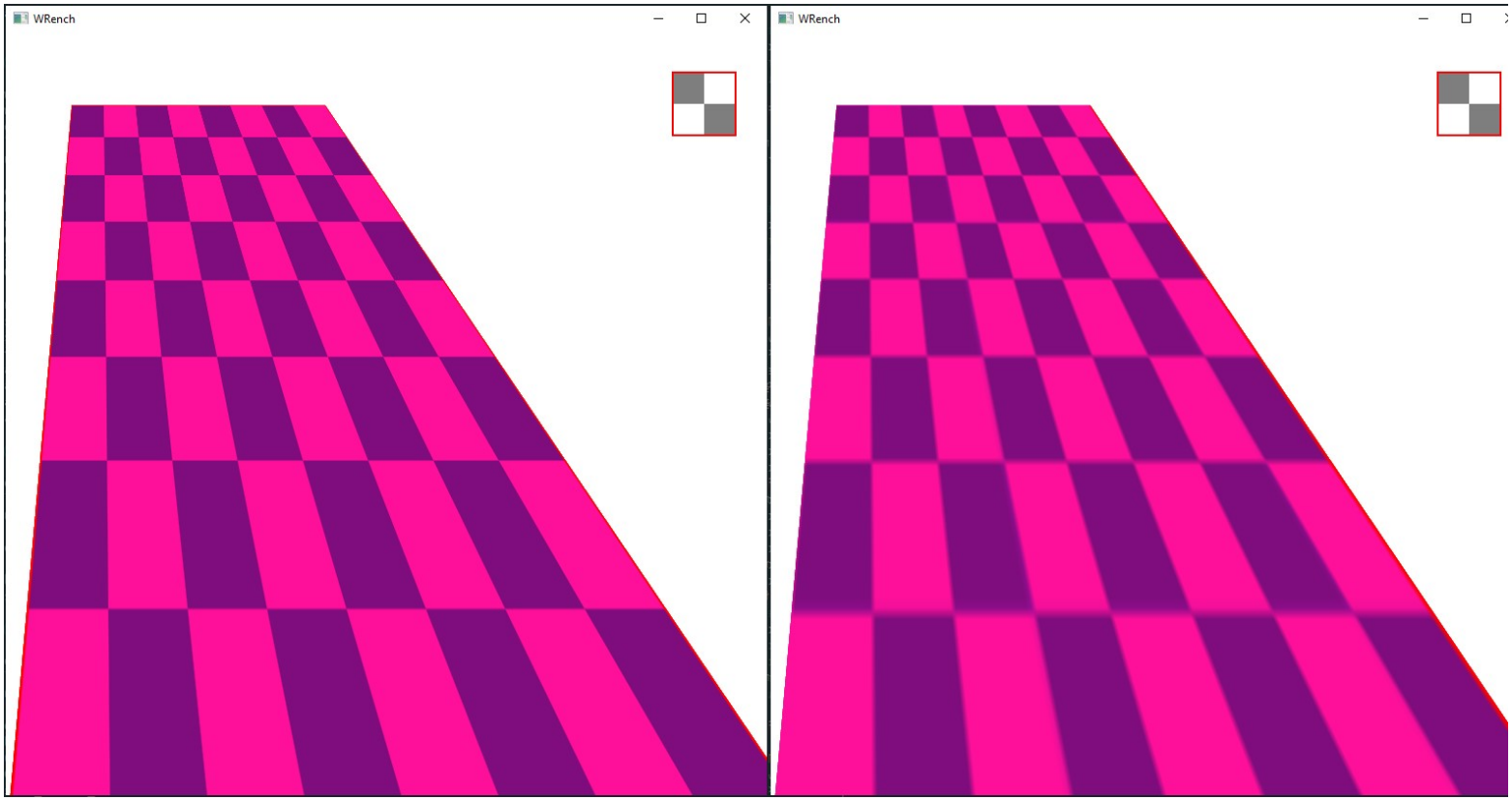

```
MAX_SURFACE_SIZE 4096

[BPE] TileCacheInstance::pre_update pic_rect Rect(866.0×816.0 at (-1.0,40.0))
[BPE] world_culling_rect Rect(1024.0×1024.0 at (0.0,0.0))
[BPE] PictureCompositeMode::TileCache::..
[BPE] establishes_raster_root
[BPE] PictureCompositeMode::ComponentTransferFilter::..
[BPE] pass 0 size (800x800)
[BPE] pass 1 size (800x800)
[BPE] framebuffer size (800x800)
```

```
[BPE] pass 2 adds task location Fixed(Rect(800×800 at (0,0)))
[BPE] pass 1 adds task location PictureCache
        { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 0 }, size: 1024×512 }
[BPE] pass 1 adds task location PictureCache
        { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 1 }, size: 1024×512 }
[BPE] pass 0 adds task location Dynamic(None, 516×516)

MAX_SURFACE_SIZE 128

[BPE] TileCacheInstance::pre_update pic_rect Rect(866.0×816.0 at (-1.0,40.0))
[BPE] world_culling_rect Rect(1024.0×1024.0 at (0.0,0.0))
[BPE] PictureCompositeMode::TileCache::..
[BPE] establishes_raster_root
[BPE] PictureCompositeMode::ComponentTransferFilter::..
[BPE] scaling factor 0.24612403
[BPE] device_rect Rect(127.0×127.0 at (9.844961,9.844961))
[BPE] pass 0 size (800x800)
[BPE] pass 1 size (800x800)
[BPE] framebuffer size (800x800)
[BPE] pass 2 adds task location Fixed(Rect(800×800 at (0,0)))
[BPE] pass 1 adds task location PictureCache
        { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 0 }, size: 1024×512 }
[BPE] pass 1 adds task location PictureCache
        { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 1 }, size: 1024×512 }
[BPE] pass 0 adds task location Dynamic(None, 128×128)
```

SVG Filter

```yaml
root:
  items:
    - type: "stacking-context"
      perspective: 100
      perspective-origin: 100 100
      items:
        - type: "stacking-context"
          filter-primitives:
          - type: flood
            color: [255, 10, 156, 1]
            color-space: srgb
          - type: blend
            in1: original
            in2: 0
            color-space: srgb
            blend-mode: darken
          transform: rotate-x(15)
          items:
            - image: checkerboard(2, 64, 8);
              bounds: [40, 40, 516, 516]
        - image: checkerboard(2, 32, 2);
```

```
            bounds: [700, 40, 68, 68]
```

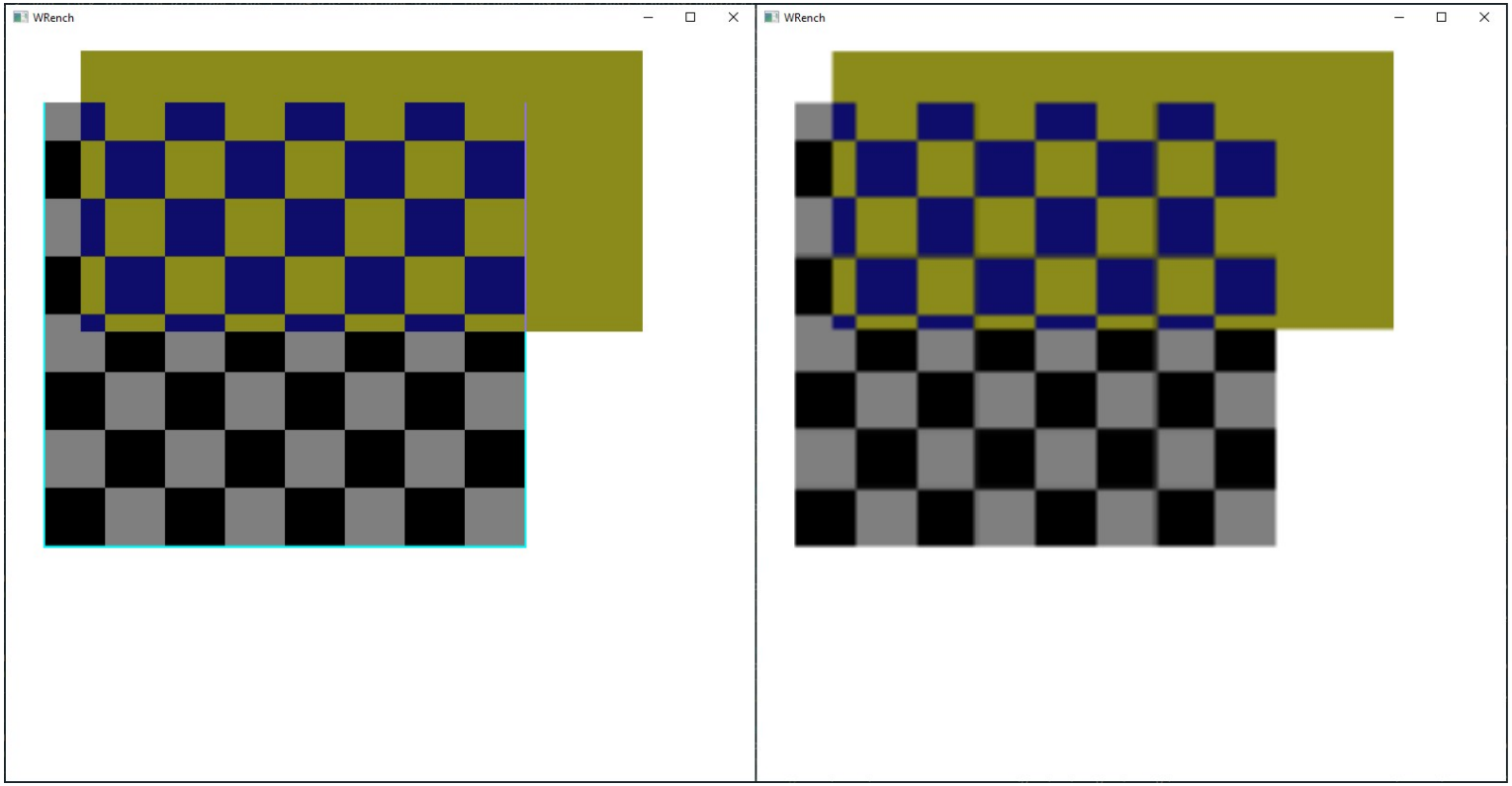



```
MAX_SURFACE_SIZE 4096

[BPE] TileCacheInstance::pre_update pic_rect Rect(866.0×816.0 at (-1.0,40.0))
[BPE] world_culling_rect Rect(1024.0×1024.0 at (0.0,0.0))
[BPE] PictureCompositeMode::TileCache::..
[BPE] establishes_raster_root
[BPE] PictureCompositeMode::SvgFilter
[BPE] with_dynamic_location size 516×516
[BPE] with_dynamic_location size 516×516
[BPE] pass 0 size (800x800)
[BPE] pass 1 size (800x800)
[BPE] pass 2 size (800x800)
[BPE] framebuffer size (800x800)
[BPE] pass 3 adds task location Fixed(Rect(800×800 at (0,0)))
[BPE] pass 2 adds task location PictureCache
        { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 0 }, size: 1024×512 }
[BPE] pass 2 adds task location PictureCache
        { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 1 }, size: 1024×512 }
[BPE] pass 0 adds task location Dynamic(None, 516×516)
```

```
[BPE] pass 0 adds task location Dynamic(None, 516×516)
[BPE] pass 1 adds task location Dynamic(None, 516×516)

MAX_SURFACE_SIZE 128

[BPE] TileCacheInstance::pre_update pic_rect Rect(866.0×816.0 at (-1.0,40.0))
[BPE] world_culling_rect Rect(1024.0×1024.0 at (0.0,0.0))
[BPE] PictureCompositeMode::TileCache::..
[BPE] establishes_raster_root
[BPE] PictureCompositeMode::SvgFilter
[BPE] scaling factor 0.24612403
[BPE] device_rect Rect(127.0×127.0 at (9.844961,9.844961))
[BPE] with_dynamic_location size 128×128
[BPE] with_dynamic_location size 128×128
[BPE] pass 0 size (800x800)
[BPE] pass 1 size (800x800)
[BPE] pass 2 size (800x800)
[BPE] framebuffer size (800x800)
[BPE] pass 3 adds task location Fixed(Rect(800×800 at (0,0)))
[BPE] pass 2 adds task location PictureCache
        { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 0 }, size: 1024×512 }
[BPE] pass 2 adds task location PictureCache
        { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 1 }, size: 1024×512 }
[BPE] pass 0 adds task location Dynamic(None, 128×128)
[BPE] pass 0 adds task location Dynamic(None, 128×128)
[BPE] pass 1 adds task location Dynamic(None, 128×128)
```

Mix Blend

```
root:
  items:
    - type: "stacking-context"
      perspective: 100
      perspective-origin: 100 100
      items:
        - type: rect
          color: [140, 140, 20, 1.0]
          bounds: 80 20 600 300
        - type: "stacking-context"
          filters: [invert(1)]
          mix-blend-mode: difference
          transform: rotate-x(15)
          items:
            - image: checkerboard(2, 64, 8);
              bounds: [40, 40, 516, 516]
```



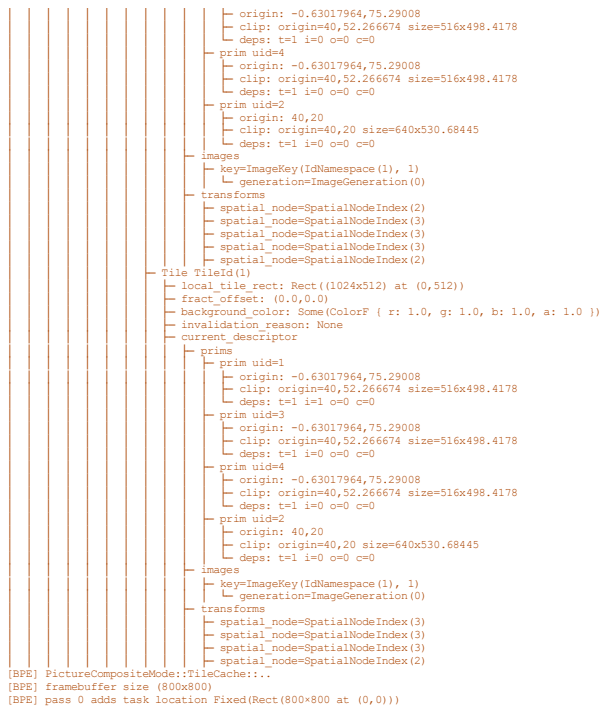
MAX_SURFACE_SIZE 4096

```
[BPE] TileCacheInstance::pre_update pic_rect Rect(640.0×531.0 at (40.0,20.0))
[BPE] world_culling_rect Rect(1024.0×1024.0 at (0.0,0.0))
[BPE] PictureCompositeMode::TileCache:...
[BPE] establishes_raster_root
[BPE] PictureCompositeMode::MixBlend../Blit_
[BPE] PictureCompositeMode::MixBlend../Blit_
[BPE] PictureCompositeMode::Filter:...
[BPE] PictureCompositeMode::RTL:Dynamic due to Filter(..) clipped Rect(516×476 at (40,75)) unclipped Rect(516.0×499.0 at (40.0,52.0))
[BPE] pass 0 size (800x800)
[BPE] pass 1 size (800x800)
[BPE] pass 2 size (800x800)
[BPE] pass 3 size (800x800)
[BPE] framebuffer size (800x800)
[BPE] pass 4 adds task location Fixed(Rect(800×800 at (0,0)))
[BPE] pass 3 adds task location PictureCache { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 0 }, size: 1024×512 }
[BPE] pass 3 adds task location PictureCache { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 1 }, size: 1024×512 }
[BPE] pass 2 adds task location Dynamic(None, 640×531)
[BPE] pass 1 adds task location Dynamic(None, 516×499)
[BPE] pass 0 adds task location Dynamic(None, 516×476)
[BPE] 'B'
[BPE] TileCacheInstance::pre_update pic_rect Rect(640.0×531.0 at (40.0,20.0))
[BPE] world_culling_rect Rect(1024.0×1024.0 at (0.0,0.0))
┌ Primitive Store
│   ├─ PictureIndex(5) at 0x2268e6d3a20
│   │   ├─ cluster_count: 1
│   │   ├─ estimated_local_rect: Rect(0.0×0.0 at (0.0,0.0))
│   │   ├─ precise_local_rect: Rect(0.0×0.0 at (0.0,0.0))
│   │   ├─ spatial_node_index: SpatialNodeIndex(0)
│   │   ├─ raster_config: None
│   │   ├─ requested_composite_mode: None
│   │   ├─ 1 Clusters
│   │   │   ├─ Cluster 0
│   │   │   │   ├─ PictureIndex(4) at 0x2268e6d3720
│   │   │   │   │   ├─ cluster_count: 1
│   │   │   │   │   ├─ estimated_local_rect: Rect(640.0×531.0 at (40.0,20.0))
│   │   │   │   │   ├─ precise_local_rect: Rect(640.0×531.0 at (40.0,20.0))
│   │   │   │   │   ├─ spatial_node_index: SpatialNodeIndex(0)
│   │   │   │   │   ├─ raster_config: Some(RasterConfig { composite_mode: TileCache, surface_index: SurfaceIndex(1), establishes_raster_root: false, task_scale: 1.0 })
│   │   │   │   │   ├─ requested_composite_mode: Some(TileCache)
│   │   │   │   │   ├─ 1 Clusters
│   │   │   │   │   │   ├─ Cluster 0
│   │   │   │   │   │   │   ├─ PictureIndex(3) at 0x2268e6d3420
│   │   │   │   │   │   │   │   ├─ cluster_count: 2
│   │   │   │   │   │   │   │   ├─ estimated_local_rect: Rect(640.0×530.68445 at (40.0,20.0))
```


precise_local_rect: Rect(640.0×530.68445 at (40.0,20.0))
spatial_node_index: SpatialNodeIndex(2)
raster_config: Some(RasterConfig { composite_mode: Blit(ISOLATE), surface_index: SurfaceIndex(2), establishes_raster_root: true, task_scale: 1.0 })
requested_composite_mode: Some(Blit(ISOLATE))
2 Clusters
Cluster 0
Rectangle { data_handle: Handle { index: 0, epoch: Epoch(2), uid: ItemUid { uid: 0 }, _marker: PhantomData }, opacity_binding_index: Index(4294967295, PhantomData), segment_instance_index: Index(4294967295, PhantomData) }
Cluster 1
PictureIndex(2) at 0x2268e6d3120
cluster_count: 1
estimated_local_rect: Rect(516.0×516.0 at (40.0,40.0))
precise_local_rect: Rect(516.0×516.0 at (40.0,40.0))
spatial_node_index: SpatialNodeIndex(3)
raster_config: Some(RasterConfig { composite_mode: MixBlend(Difference), surface_index: SurfaceIndex(3), establishes_raster_root: false, task_scale: 1.0 })
requested_composite_mode: Some(MixBlend(Difference))
1 Clusters
Cluster 0
PictureIndex(1) at 0x2268e6d2e20
cluster_count: 1
estimated_local_rect: Rect(516.0×516.0 at (40.0,40.0))
precise_local_rect: Rect(516.0×516.0 at (40.0,40.0))
spatial_node_index: SpatialNodeIndex(3)
raster_config: Some(RasterConfig { composite_mode: Filter(Invert(1.0)), surface_index: SurfaceIndex(4), establishes_raster_root: false, task_scale: 1.0 })
requested_composite_mode: Some(Filter(Invert(1.0)))
1 Clusters
Cluster 0
PictureIndex(0) at 0x2268e6d2b20
cluster_count: 1
estimated_local_rect: Rect(0.0×0.0 at (0.0,0.0))
precise_local_rect: Rect(0.0×0.0 at (0.0,0.0))
spatial_node_index: SpatialNodeIndex(3)
raster_config: None
requested_composite_mode: None
1 Clusters
Cluster 0
Image { data_handle: Handle { index: 0, epoch: Epoch(2), uid: ItemUid { uid: 1 }, _marker: PhantomData }, image_instance_index: Index(0, PhantomData) }
TileCache
Slice 0 at 0x2268e7e61a0
fract_offset: (0.0,0.0)
background_color: Some(ColorF { r: 1.0, g: 1.0, b: 1.0, a: 1.0 })
local_rect: Rect(640.0×531.0 at (40.0,20.0))
Tile TileId(0)
local_tile_rect: Rect((1024x512) at (0,0))
fract_offset: (0.0,0.0)
background_color: Some(ColorF { r: 1.0, g: 1.0, b: 1.0, a: 1.0 })
invalidation_reason: None
current_descriptor
prims
prim uid=0
origin: 80,20
clip: origin=80,20 size=600x300
deps: t=1 i=0 o=0 c=0
prim uid=1
origin: -0.63017964,75.29008
clip: origin=40,52.266674 size=516x498.4178
deps: t=1 i=1 o=0 c=0
prim uid=3
origin: -0.63017964,75.29008
clip: origin=40,52.266674 size=516x498.4178
deps: t=1 i=0 o=0 c=0
prim uid=4
origin: -0.63017964,75.29008
clip: origin=40,52.266674 size=516x498.4178
deps: t=1 i=0 o=0 c=0
prim uid=2
origin: 40,20
clip: origin=40,20 size=640x530.68445
deps: t=1 i=0 o=0 c=0
images
key=ImageKey(IdNamespace(1), 1)
generation=ImageGeneration(0)
transforms
spatial_node=SpatialNodeIndex(2)
spatial_node=SpatialNodeIndex(3)
spatial_node=SpatialNodeIndex(3)
spatial_node=SpatialNodeIndex(3)
spatial_node=SpatialNodeIndex(2)
Tile TileId(1)
local_tile_rect: Rect((1024x512) at (0,512))
fract_offset: (0.0,0.0)
background_color: Some(ColorF { r: 1.0, g: 1.0, b: 1.0, a: 1.0 })
invalidation_reason: None
current_descriptor
prims
prim uid=1
origin: -0.63017964,75.29008
clip: origin=40,52.266674 size=516x498.4178
deps: t=1 i=1 o=0 c=0
prim uid=3
origin: -0.63017964,75.29008
clip: origin=40,52.266674 size=516x498.4178
deps: t=1 i=0 o=0 c=0
prim uid=4
origin: -0.63017964,75.29008
clip: origin=40,52.266674 size=516x498.4178
deps: t=1 i=0 o=0 c=0
prim uid=2
origin: 40,20
clip: origin=40,20 size=640x530.68445
deps: t=1 i=0 o=0 c=0
images
key=ImageKey(IdNamespace(1), 1)
generation=ImageGeneration(0)
transforms
spatial_node=SpatialNodeIndex(3)
spatial_node=SpatialNodeIndex(3)
spatial_node=SpatialNodeIndex(3)
spatial_node=SpatialNodeIndex(2)
[BPE] PictureCompositeMode::TileCache::..
[BPE] framebuffer size (800x800)
[BPE] pass 0 adds task location Fixed(Rect(800×800 at (0,0)))

MAX_SURFACE_SIZE 128

```
[BPE] TileCacheInstance::pre_update pic_rect Rect(640.0×531.0 at (40.0,20.0))
[BPE] world_culling_rect Rect(1024.0×1024.0 at (0.0,0.0))
[BPE] PictureCompositeMode::TileCache::..
[BPE] establishes_raster_root
[BPE] PictureCompositeMode::MixBlend../Blit_
[BPE] scaling factor 0.1984375
[BPE] device_rect Rect(127.0×105.37031 at (7.9375,3.96875))
[BPE] PictureCompositeMode::MixBlend../Blit_
[BPE] PictureCompositeMode::Filter::..
[BPE] PictureCompositeMode::RTL:Dynamic due to Filter(..) clipped Rect(516×476 at (40,75)) unclipped Rect(516.0×499.0 at (40.0,52.0))
[BPE] pass 0 size (800x800)
[BPE] pass 1 size (800x800)
[BPE] pass 2 size (800x800)
[BPE] pass 3 size (800x800)
[BPE] framebuffer size (800x800)
[BPE] pass 4 adds task location Fixed(Rect(800×800 at (0,0)))
[BPE] pass 3 adds task location PictureCache { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 0 }, size: 1024×512 }
[BPE] pass 3 adds task location PictureCache { surface: TextureCache { texture: TextureCache(CacheTextureId(4), Rgba), layer: 1 }, size: 1024×512 }
[BPE] pass 2 adds task location Dynamic(None, 128×107)
[BPE] pass 1 adds task location Dynamic(None, 516×499)
[BPE] pass 0 adds task location Dynamic(None, 516×476)
[BPE] 'B'
[BPE] TileCacheInstance::pre_update pic_rect Rect(640.0×531.0 at (40.0,20.0))
[BPE] world_culling_rect Rect(1024.0×1024.0 at (0.0,0.0))
┌ Primitive Store
  ├ PictureIndex(5) at 0x1882ede3360
    ├ cluster_count: 1
    ├ estimated_local_rect: Rect(0.0×0.0 at (0.0,0.0))
    ├ precise_local_rect: Rect(0.0×0.0 at (0.0,0.0))
    ├ spatial_node_index: SpatialNodeIndex(0)
    ├ raster_config: None
    ├ requested_composite_mode: None
    ├ 1 Clusters
      ├ Cluster 0
        ├ PictureIndex(4) at 0x1882ede3060
          ├ cluster_count: 1
          ├ estimated_local_rect: Rect(640.0×531.0 at (40.0,20.0))
          ├ precise_local_rect: Rect(640.0×531.0 at (40.0,20.0))
          ├ spatial_node_index: SpatialNodeIndex(0)
          ├ raster_config: Some(RasterConfig { composite_mode: TileCache, surface_index: SurfaceIndex(1), establishes_raster_root: false, task_scale: 1.0 })
          ├ requested_composite_mode: Some(TileCache)
          ├ 1 Clusters
            ├ Cluster 0
              ├ PictureIndex(3) at 0x1882ede2d60
                ├ cluster_count: 2
                ├ estimated_local_rect: Rect(640.0×530.68445 at (40.0,20.0))
                ├ precise_local_rect: Rect(640.0×530.68445 at (40.0,20.0))
                ├ spatial_node_index: SpatialNodeIndex(2)
                ├ raster_config: Some(RasterConfig { composite_mode: Blit(ISOLATE), surface_index: SurfaceIndex(2), establishes_raster_root: true, task_scale: 1.0 })
                ├ requested_composite_mode: Some(Blit(ISOLATE))
                ├ 2 Clusters
                  ├ Cluster 0
                    └ Rectangle { data_handle: Handle { index: 0, epoch: Epoch(2), uid: ItemUid { uid: 0 }, _marker: PhantomData }, opacity_binding_index: Index(4294967295, PhantomData), segment_instance_index: Index(4294967295, PhantomData) }
                  ├ Cluster 1
                    ├ PictureIndex(2) at 0x1882ede2a60
                      ├ cluster_count: 1
                      ├ estimated_local_rect: Rect(516.0×516.0 at (40.0,40.0))
                      ├ precise_local_rect: Rect(516.0×516.0 at (40.0,40.0))
                      ├ spatial_node_index: SpatialNodeIndex(3)
                      ├ raster_config: Some(RasterConfig { composite_mode: MixBlend(Difference), surface_index: SurfaceIndex(3), establishes_raster_root: false, task_scale: 1.0 })
                      ├ requested_composite_mode: Some(MixBlend(Difference))
                      ├ 1 Clusters
                        ├ Cluster 0
                          ├ PictureIndex(1) at 0x1882ede2760
                            ├ cluster_count: 1
                            ├ estimated_local_rect: Rect(516.0×516.0 at (40.0,40.0))
                            ├ precise_local_rect: Rect(516.0×516.0 at (40.0,40.0))
                            ├ spatial_node_index: SpatialNodeIndex(3)
                            ├ raster_config: Some(RasterConfig { composite_mode: Filter(Invert(1.0)), surface_index: SurfaceIndex(4), establishes_raster_root: false, task_scale: 1.0 })
                            ├ requested_composite_mode: Some(Filter(Invert(1.0)))
                            ├ 1 Clusters
                              ├ Cluster 0
                                ├ PictureIndex(0) at 0x1882ede2460
                                  ├ cluster_count: 1
                                  ├ estimated_local_rect: Rect(0.0×0.0 at (0.0,0.0))
                                  ├ precise_local_rect: Rect(0.0×0.0 at (0.0,0.0))
                                  ├ spatial_node_index: SpatialNodeIndex(3)
                                  ├ raster_config: None
                                  ├ requested_composite_mode: None
                                  ├ 1 Clusters
                                    ├ Cluster 0
                                      └ Image { data_handle: Handle { index: 0, epoch: Epoch(2), uid: ItemUid { uid: 1 }, _marker: PhantomData }, image_instance_index: Index(0, PhantomData) }
          ├ TileCache
            ├ Slice 0 at 0x1882ee7fa60
              ├ fract_offset: (0.0,0.0)
              ├ background_color: Some(ColorF { r: 1.0, g: 1.0, b: 1.0, a: 1.0 })
              ├ local_rect: Rect(640.0×531.0 at (40.0,20.0))
              ├ Tile TileId(0)
                ├ local_tile_rect: Rect((1024x512) at (0,0))
                ├ fract_offset: (0.0,0.0)
                ├ background_color: Some(ColorF { r: 1.0, g: 1.0, b: 1.0, a: 1.0 })
                ├ invalidation_reason: None
                ├ current_descriptor
                  ├ prims
                    ├ prim uid=0
                      ├ origin: 80,20
                      ├ clip: origin=80,20 size=600x300
                      └ deps: t=1 i=0 o=0 c=0
                    ├ prim uid=1
                      ├ origin: -0.63017964,75.29008
                      ├ clip: origin=40,52.266674 size=516x498.4178
                      └ deps: t=1 i=1 o=0 c=0
                    ├ prim uid=3
```



Diff:

```
[BPE] scaling factor 0.1984375
[BPE] device_rect Rect(127.0×105.37031 at (7.9375,3.96875))
...
[BPE] pass 2 adds task location Dynamic(None, 128×107)
```