

---

Subject: Re: BufferPainter::Clear() optimization  
Posted by Tom1 on Wed, 20 May 2020 10:52:53 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

mirek wrote on Wed, 20 May 2020 13:23 OK, after retesting, I think it might be at most 3% faster. Looking at fillers, I think there is much more time spent in AlphaBlend function - even if it is just for segment start/end pixels. Perhaps that one should be SSE2 optimized?

Mirek

Hi,

My SSE2 battery is now 'discharged' for a while.... Need to recharge before next use.

I also did some testing on span filler with memcpy. This is based on using IMAGE\_OPAQUE of the image being rendered. It does improve the speed somewhat, but the edges cause a problem since the edge is alpha blended even if FILL\_FAST is specified. So, this needs some reconsideration and better knowledge on the Painter internals (i.e. beyond my level...):

BufferPainter.h:

```
struct SpanSource {
    int kind;
    SpanSource(){
        kind = IMAGE_OPAQUE;
    }
    virtual void Get(RGBA *span, int x, int y, unsigned len) = 0;
    virtual ~SpanSource() {}
};
```

Fillers.cpp:

```
void SpanFiller::Render(int val, int len)
{
    if(val == 0) {
        t += len;
        s += len;
        return;
    }
    if(alpha != 256)
        val = alpha * val >> 8;

    if(val == 256) {
        if(ss->kind==IMAGE_OPAQUE) memcpy(t,s,len*sizeof(RGBA)); // apex_memcpy() would be
even faster
    }
    else{
        for(int i = 0; i < len; i++) {
            if(s[i].a == 255)
```

```
    t[i] = s[i];
  else
    AlphaBlend(t[i], s[i]);
}
}
t += len;
s += len;
}
else {
  const RGBA *e = t + len;
  while(t < e)
    AlphaBlendCover8(*t++, *s++, val);
}
}
```

Painter/Image.cpp:

```
struct PainterImageSpan : SpanSource, PainterImageSpanData {
  LinearInterpolator interpolator;
```

```
  PainterImageSpan(const PainterImageSpanData& f)
  : PainterImageSpanData(f) {
    interpolator.Set(xform);
    kind = image.GetKindNoScan(); // Add this
  }
```

Best regards,

Tom

---