
Subject: Re: Painter Fill with Image MSC14x64 performance issue

Posted by [Tom1](#) on Wed, 11 Jan 2017 09:01:32 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Mirek,

Thanks for pointing out the correct location for investigation:

With a few changes, I have succeeded to more than double the performance: UHD sized FAST_FILL render was dropped from 97 ms to 35 ms when using MSC14x64. For MSC14 the improvement was observable with a drop from 59 ms to 40 ms.

First BufferPainter.h in class LinearInterpolator I have inlined 'int Dda2::Get()' and 'Point Get()' functions (and removed them from Interpolator.cpp):

```
class LinearInterpolator {
    struct Dda2 {
        int count, lift, rem, mod, p;

        void Set(int a, int b, int len);
    // int Get();
        int Get()
        {
            int pp = p;
            mod += rem;
            p += lift;
            if(mod > 0) {
                mod -= count;
                p++;
            }
            return pp;
        }
    };

    Xform2D xform;
    Dda2 ddax, dday;

    static int Q8(double x) { return int(256 * x + 0.5); }

public:
    void Set(const Xform2D& m) { xform = m; }

    void Begin(int x, int y, int len);
    // Point Get();
    Point Get()
    {
        return Point(ddax.Get(), dday.Get());
    }
}
```

```
};
```

Second Image.cpp in struct PainterImageSpan at the beginning of 'virtual void Get(RGBA *span, int x, int y, unsigned len)' I have added optimized code for FAST_FILL without effect flags. This mostly improves MSC14x64 results:

```
virtual void Get(RGBA *span, int x, int y, unsigned len)
{
    interpolator.Begin(x, y, len);
    fixed = hstyle && vstyle;

    if((hstyle|vstyle)==0 && fast){
        while(len--){
            Point l = interpolator.Get() >> 8;
            if(l.x > 0 && l.x < maxx && l.y > 0 && l.y < maxy) *span = Pixel(l.x, l.y);
            else if(style == 0 && (l.x < -1 || l.x > cx || l.y < -1 || l.y > cy)) *span = RGBAZero();
            else *span = GetPixel(l.x, l.y);
            ++span;
        }
        return;
    }

    while(len--) {
        Point h = interpolator.Get();
        // h -= 128;
        Point l = h >> 8;
        ...
    }
}
```

Finally, some more milliseconds can be squeezed out by changing SpanFiller::Render(int val, int len) in Fillers.cpp as follows. Using 'for' instead of 'while' seems to have positive effect mostly on MSC14x64:

```
void SpanFiller::Render(int val, int len)
{
    if(val == 0) {
        t += len;
        s += len;
        return;
    }
    const RGBA *e = t + len;
    if(alpha != 256)
        val = alpha * val >> 8;
    if(val == 256)
        for(int i=0;i<len;i++) if(s[i].a==255) t[i]=s[i]; else AlphaBlend(t[i], s[i]);
    /* while(t < e) {
        if(s->a == 255)
```

```
    *t++ = *s++;  
    else  
        AlphaBlend(*t++, *s++);  
}  
*/ else  
    while(t < e)  
        AlphaBlendCover8(*t++, *s++, val);  
}
```

Please evaluate the changes and commit if you agree with me.

Best regards,

Tom

UPDATE: There was a slight error in the positioning of "fixed = hstyle && vstyle;" potentially causing crash. Fixed above.

Another UPDATE: There was an extra "LinearInterpolator::" for "Point LinearInterpolator::Get()" above. For Windows it was ok, but CLANG found it. Sorry for that one too. Fixed now.