

2025-03-28-vglare Parametric 2d Gaussian Fourier Series

진행 사항

- rex 상에서 parametric pre-compute 구현

K 개의 2d gaussian sample을 세팅. coefficient table을 만드는 것과 픽셀에서 K개 gaussian 각 계산 후 합치는 과정 구현 후 rendering 속도 차이를 비교한다.

구현 목적

k개의 function 연산을 fourier-series기반의 series로 분해하여, pixel position에 independent한 coefficients만 pre-compute한다. 실제 렌더링 프레임에서는 series의 정확도(order)에만 dependent한, constant time에 연산할 수 있도록 한다.

pre-compute파트 즉, initial_update()에서 coefficients를 연산하여 buffer에 저장한다. buffer를 ssbo 형식으로 shader에 전달하여, fragment shader에서는 $O(1)$ 시간으로 function 연산을 진행한다.

성능 비교를 위해 pre-compute를 하지 않고, shader에서 k개의 function 연산을 진행하는 부분을 구현하여 비교했다.

Implementation

- userdata에 coefficients 저장하는 table과 SSBO setting

```
gl::Buffer* COEF = nullptr;  
vector<float> A = vector<float>(N * M, 0.0);
```

- initial_update 파트에서 각 ghost에 대한 x_k, y_k 를 포함한 fourier coefficient를 pre-compute.
각 계수를 A, B, C, D table에 저장

```
// initial_update()  
for m in [0, M-1]  
    for n in [0, N-1]  
        a_n, b_m = gaussian_coeff(n, m, sigma)  
        kappa = (n==0 && m==0) ? 1.0 : ((n==0 || m==0) ? 2.0 : 4.0);  
        coef = kappa * a_n * b_m / (T * T)  
        for k in [0, K-1]  
            x_k, y_k = ghost_means[k]  
            A[m, n] += coef * cos(n * omega * x_k) * cos(m * omega * y_k)  
            B[m, n] += coef * cos(n * omega * x_k) * sin(m * omega * y_k)  
            C[m, n] += coef * sin(n * omega * x_k) * cos(m * omega * y_k)  
            D[m, n] += coef * sin(n * omega * x_k) * sin(m * omega * y_k)
```

- A, B, C, D와 ghost의 $\text{mean}(x_k, y_k)$ 값을 SSBO에 저장한다. b_parametric_compute에 따라
shader에서 parametric compute와 non-parametric compute 수행

Benchmark Result

1. # of ghost $K = 5$, fourier degree $N = 10$, $M = 10$

1. parametric case :

FPS 4000~5000

VGlare_FSD2.draw_glare = 0.080 ms, mean GL time = 0.207ms



2. non-parametric case :

FPS 5000~6000

VGlare_FSD2.draw_glare = 0.014ms, mean GL time = 0.161ms

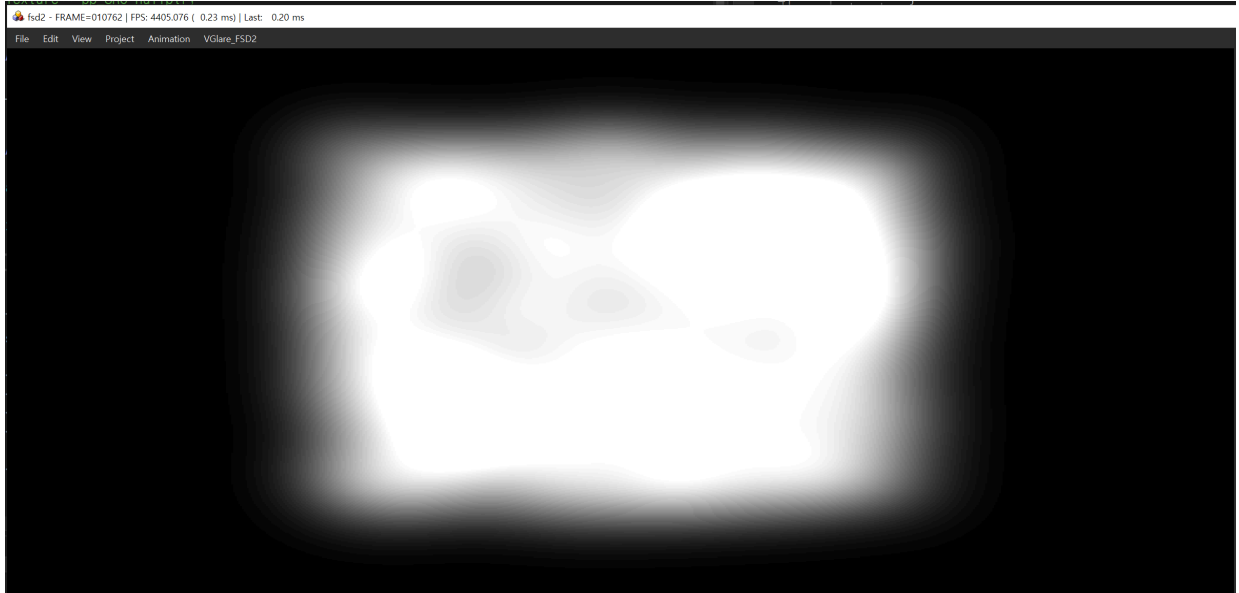


2. # of ghost $K = 1000$, fourier degree $N = 10$, $M = 10$, scaling = 0.03

1. parametric case :

FPS 4000~5000

VGlare_FSD2.draw_glare = 0.082 ms, mean GL time = 0.207ms



2. non-parametric case :

FPS ± 2000

VGlare_FSD2.draw_glare = 0.311ms, mean GL time = 0.442ms

