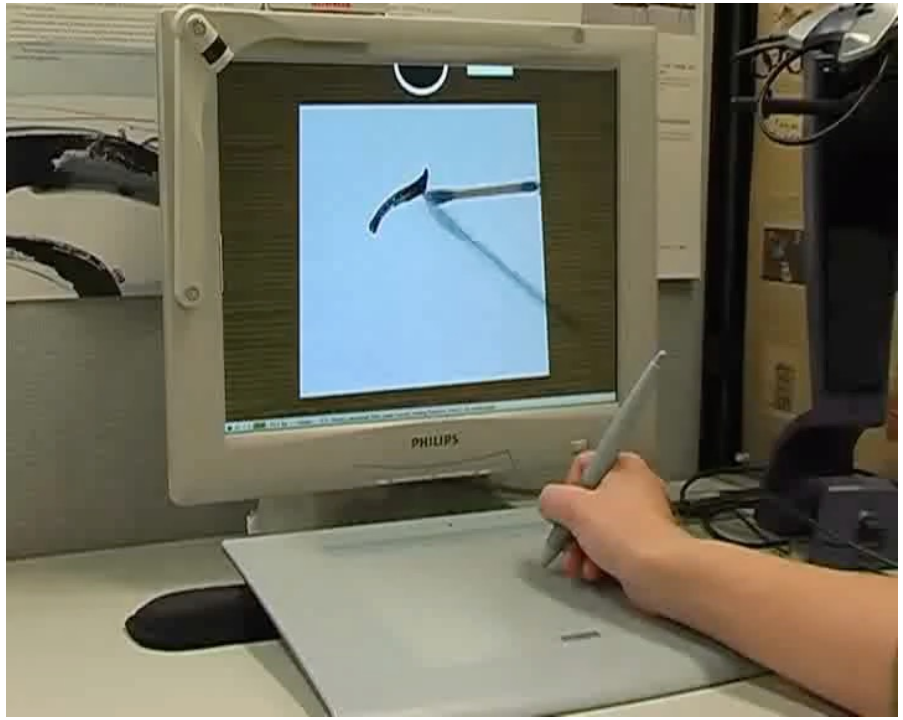
太极墨戏

Speaker：Vineyo

Background

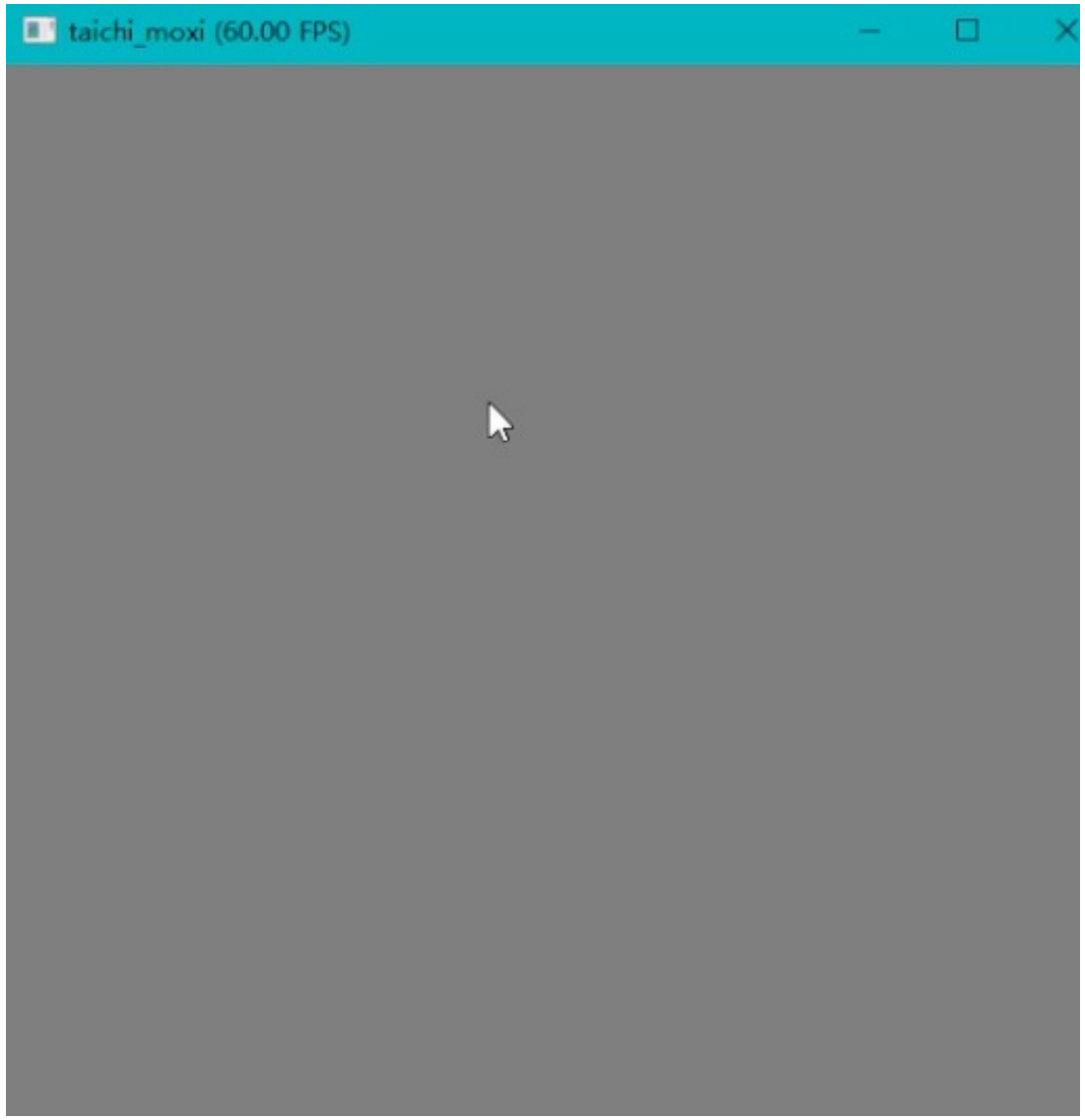
- Original paper: **MoXi: Real-Time Ink Dispersion in Absorbent Paper** (https://dl.acm.org/doi/10.1145/1073204.1073221)



2005

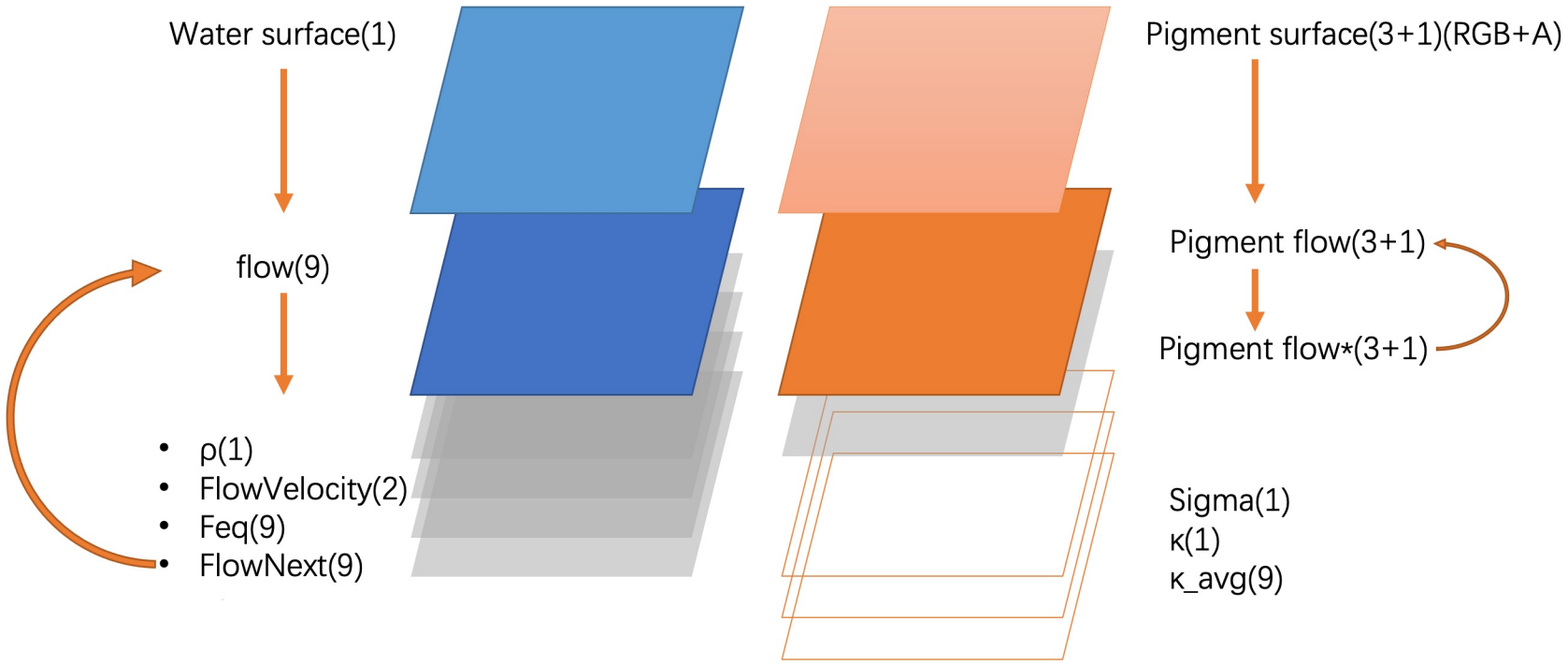
Today

My result



Overview

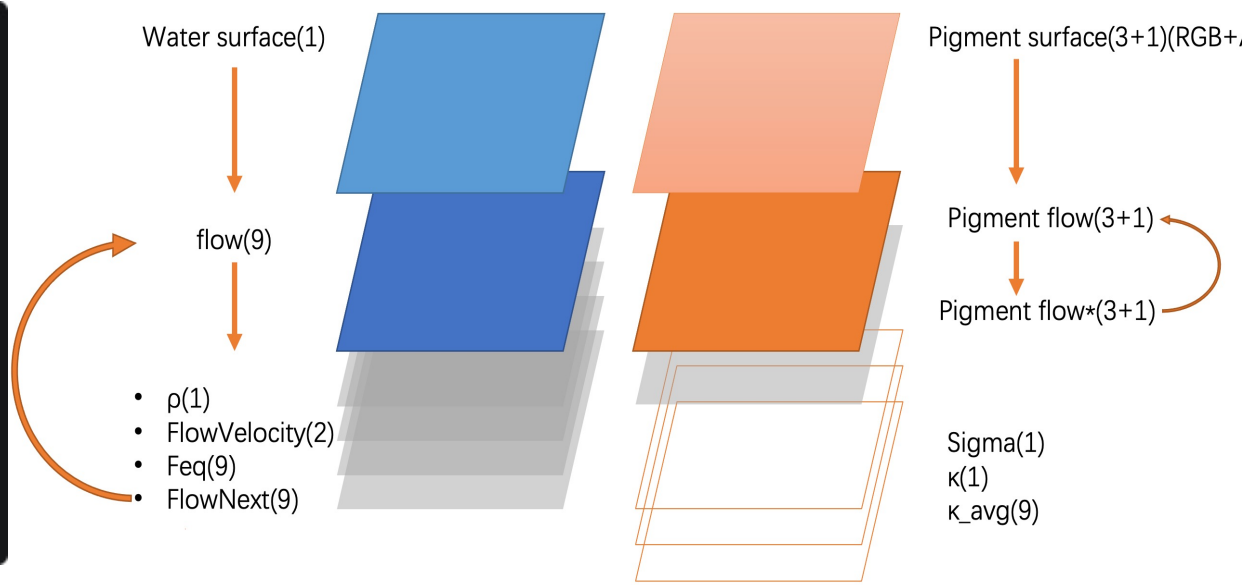
```
while Ture:
  drawing_on_surface()
  water_pigment_surface_to_flow()
  update_rho()
  update_Feq()
  update_Flow_Next()
  update_flow()
  update_Pf_star()
  update_Pf()
  update_k_for_pinning()
  update_kappa_avg()
```



Data layout

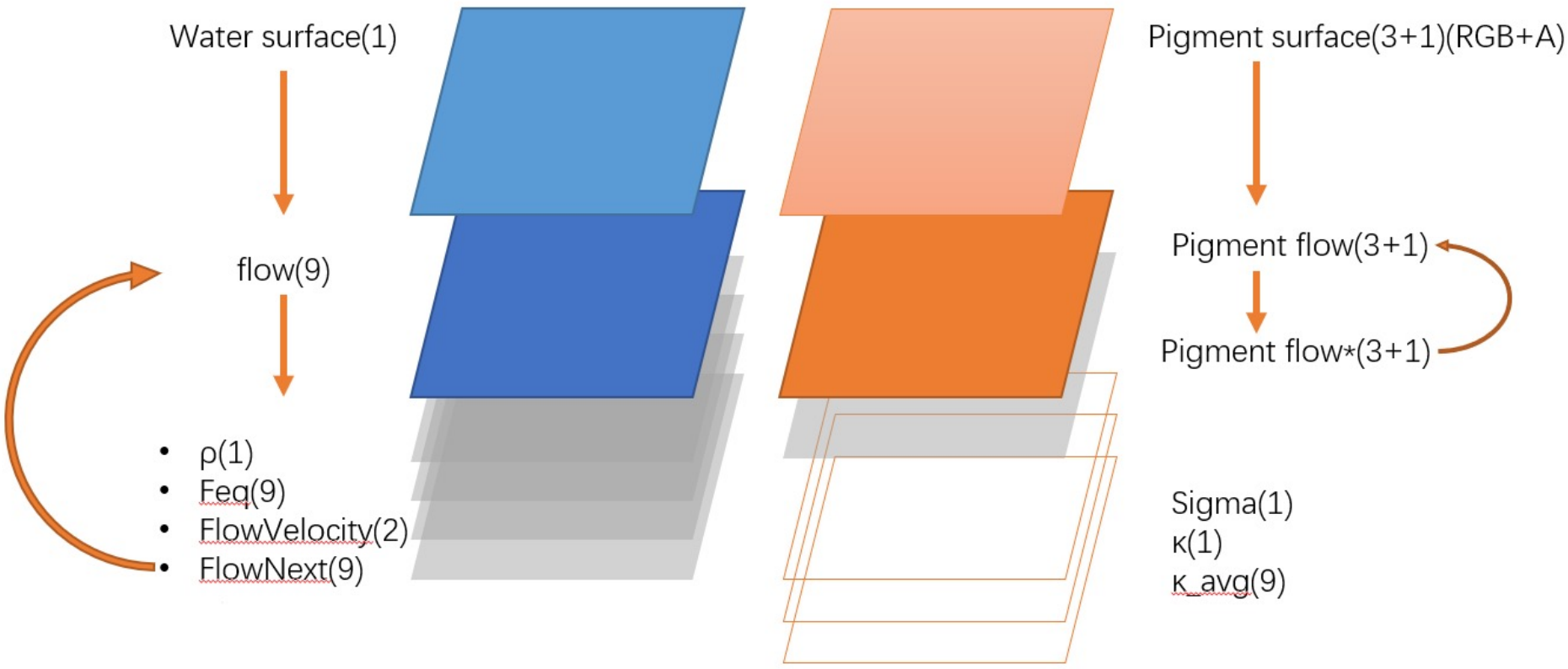
```
self.s0 = ti.root
self.s1 = self.s0.pointer(ti.ij, 16)
self.s2 = self.s1.dense(ti.ij, 8)
self.s3 = self.s2.dense(ti.ij, 4)

self.s3.place(self.Pigment_flow_c, self.Pigment_flow_a)
self.s3.place(self.Pigment_flow_star_c, self.Pigment_flow_star_a)
self.s3.place(self.Flow,self.Feq,self.FlowNext)
self.s3.place(self.rho)
self.s3.place(self.FlowVelocity)
```

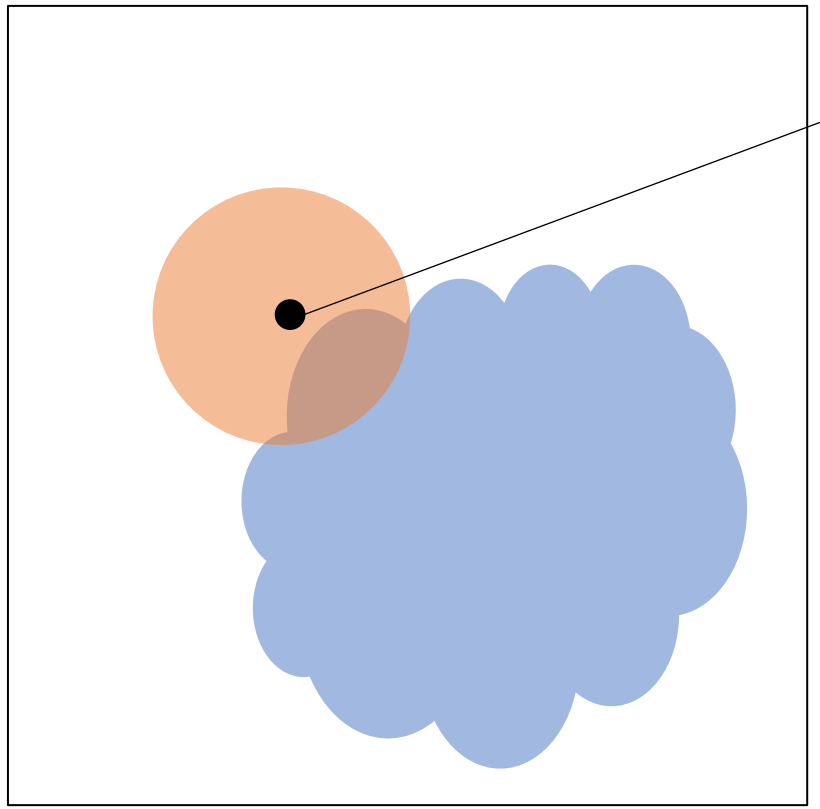


Overview

```
while Ture:
  drawing_on_surface()
  water_pigment_surface_to_flow()
  update_rho()
  update_Feq()
  update_Flow_Next()
  update_flow()
  update_Pf_star()
  update_Pf()
  update_k_for_pinning()
  update_kappa_avg()
```



Drawing on surface



If **pixel** inside the circle:

Water surface:

$$water_{surface}[P] \mathrel{+}= max(1 - \rho[P]/\lambda, m)$$

where $0.3 \leqslant \lambda \leqslant 1$, m=0.1

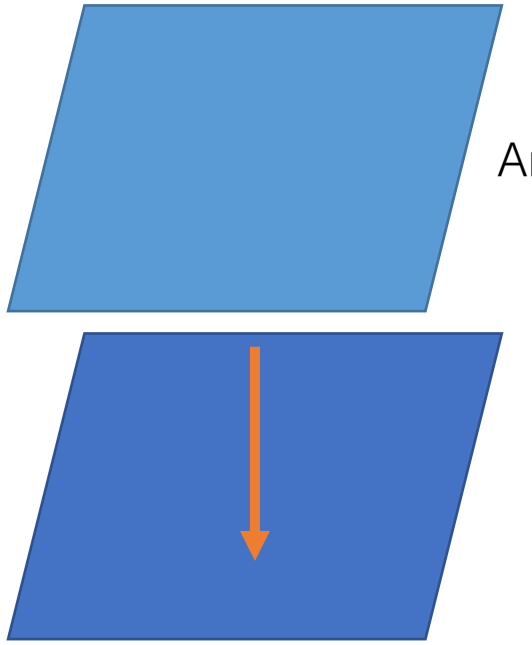
Pigment surface:

$$pigment_{surface_c}[P] = color_c$$
$$pigment_{surface_a}[P] = color_a$$

(c: RGB, a: Alpha)

Water and pigment supply

Water surface(1)



Flow(9)

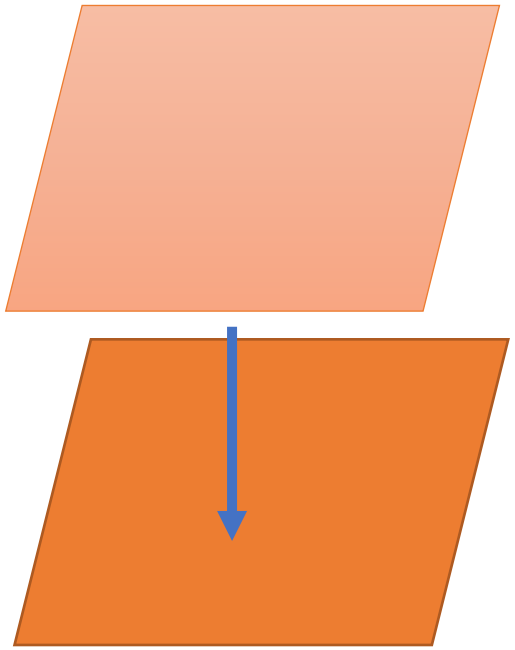
Amount of water supply to flow layer:

$$\varphi = clamp(\mathrm{W_s}, 0{,}1 - \rho)$$

$$\frac{\varphi}{W_s} = \frac{a}{P_{s_a}}$$

Where a is the amount of pigment flowing layer

Pigment surface(3+1)(RGB+A)



Pigment flow(3+1)

Water and pigment supply to flow layer like sources.

Little trick

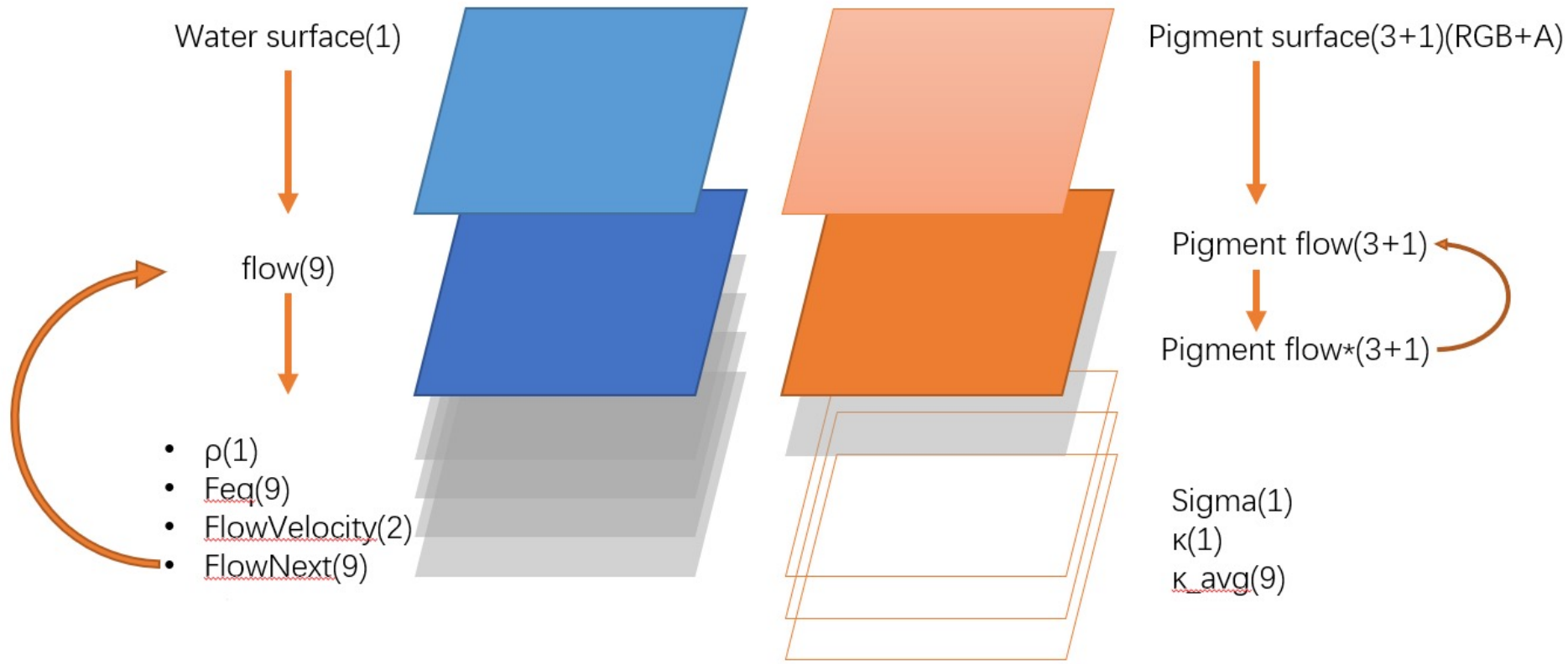
Pigment surface(3+1)(RGB+A)

```
1  self.Pigment_flow_c = ti.Vector.field(3, dtype=float)
2  self.Pigment_flow_a = ti.field(dtype=float)
3 self.s3.place(self.Pigment_flow_c, self.Pigment_flow_a)
```

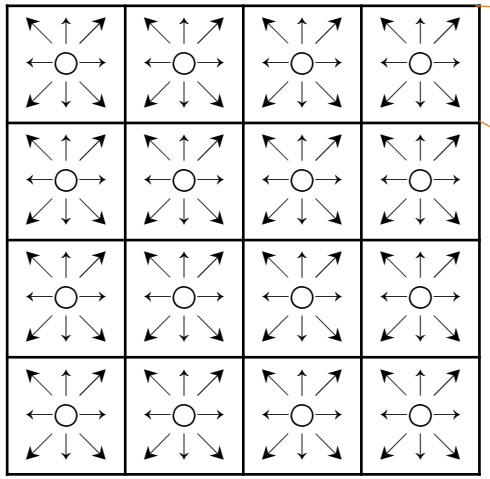
```
1 Pigment_flow_c[P] = (Pigment_flow_c[P] * Pigment_flow_a[P] + Pigment_Surface_c[P]*b) / (Pigment_flow_a[P]+b)
2 Pigment_flow_a[P] = tg.scalar.clamp(Pigment_flow_a[P]+b, 0, 1)
```

Overview

```
1 while Ture:
2   drawing_on_surface()
3   water_pigment_surface_to_flow()
4   update_rho()
5   update_Feq()
6   update_Flow_Next()
7   update_flow()
8   update_Pf_star()
9   update_Pf()
10  update_k_for_pinning()
11  update_kappa_avg()
```



Structure of flow and some basic variables



6	1	5
2	0	4
7	3	8

$f[P]$
0
1
2
3
4
5
6
7
8

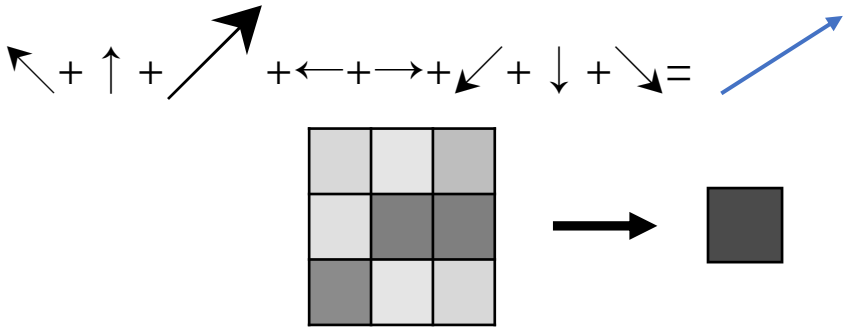
$\mathbf{e}$
[0, 0]
[0, 1]
[-1, 0]
[0, -1]
[1, 0]
[1, 1]
[-1, 1]
[-1, -1]
[1, -1]

Some derivatives of flow

$$\textbf{FlowVelocity: } \boldsymbol{u} = \sum_{i=1}^{8} f_i * \mathbf{e_i}$$

$$\textbf{Water density: } \rho = \sum_{i=0}^{8} f_i * 0.995$$

(where 0.995 for evaporation)



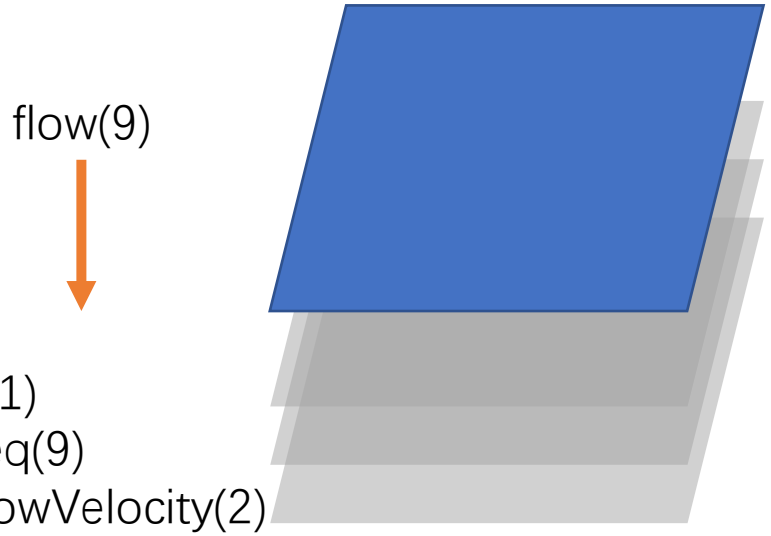
$$f_i^{(eq)} = w_i\{\rho + \psi[3\mathbf{e_i} \cdot \boldsymbol{u} + 4.5(\mathbf{e_i} \cdot \boldsymbol{u})^2 - 1.5\boldsymbol{u} \cdot \boldsymbol{u}]\}$$

where

w_0 =4/9, $w_{1:4}$ =1/9, $w_{5:8}$ =1/36

$\psi = \mathrm{smoothstep}(\rho[\mathrm{P}], 0, \alpha)$

$0.2 \leqslant \alpha \leqslant 0.5$ (make the flow conserves)

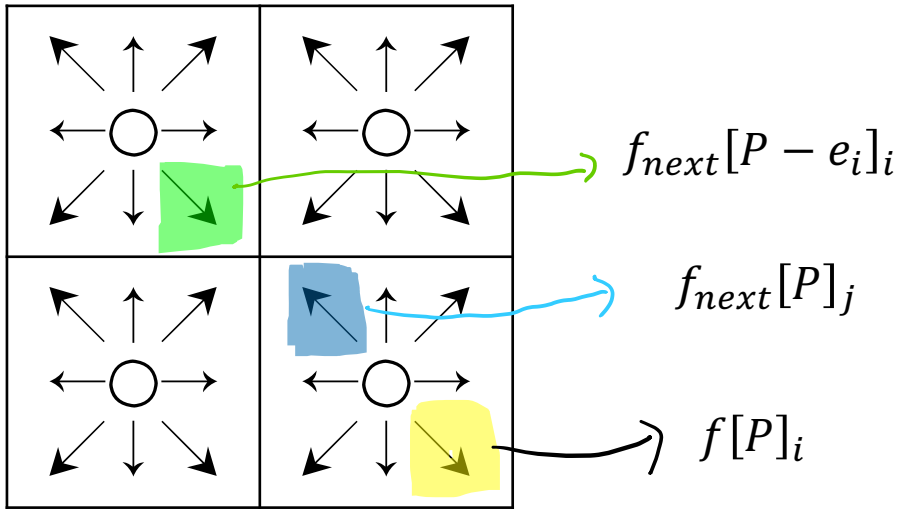


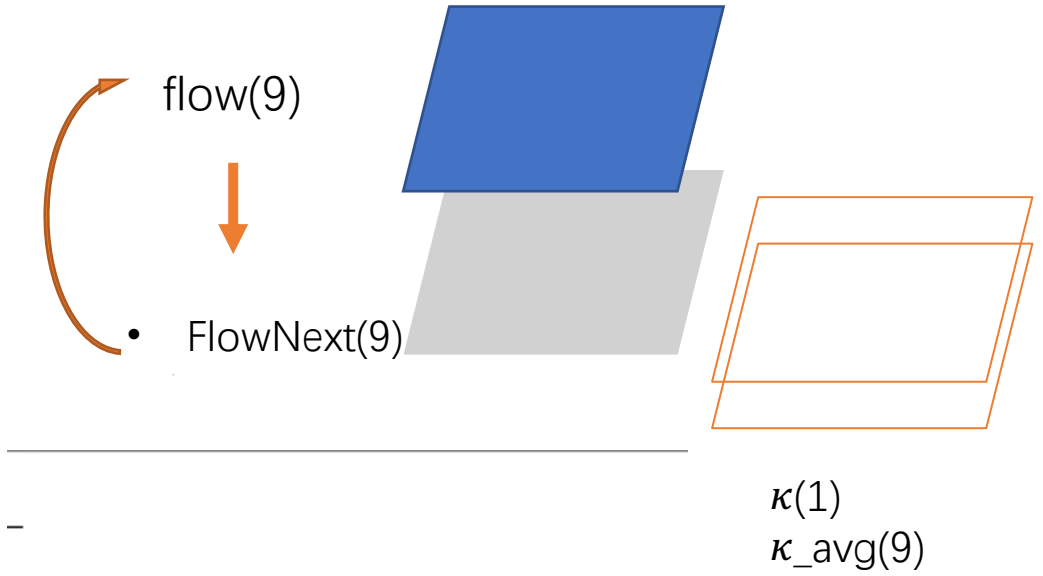
Update flow

$$f_{next}[P]_i = mix(f_i^{(eq)}[P - e_i], f[P - e_i], \omega) \qquad \omega \approx 0.5$$

$$f[P]_i = mix(f_{next}[P - e_i]_i, f_{next}[P]_j, \bar{\kappa}_i)$$

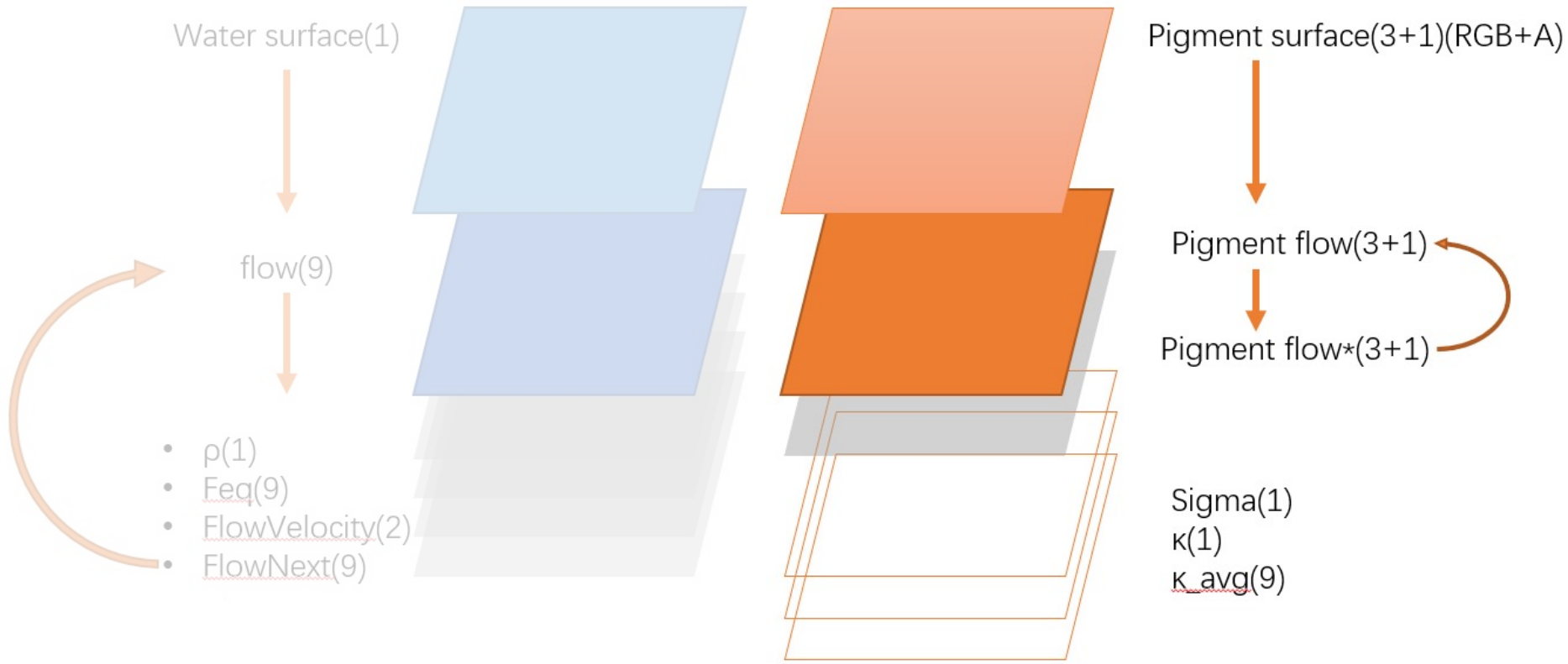
j: opposite index of i
κ: paper texture, block advection
$\bar{\kappa}$: average of two neighbor κ





Overview

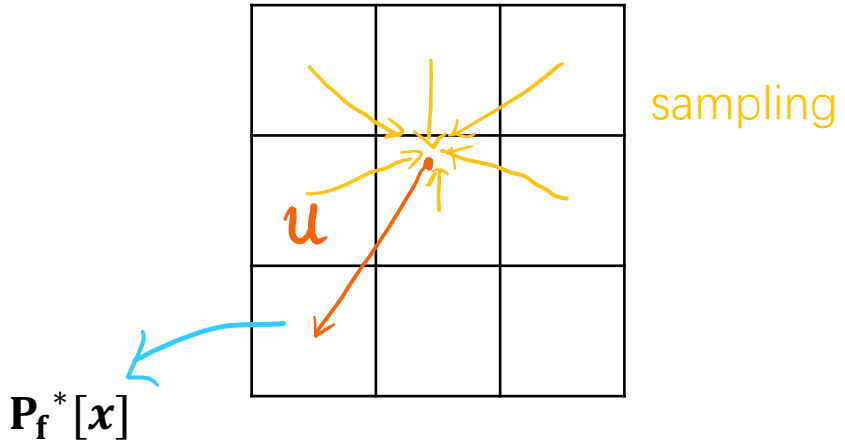
```
while Ture:
  drawing_on_surface()
  water_pigment_surface_to_flow()
  update_rho()
  update_Feq()
  update_Flow_Next()
  update_flow()
  update_Pf_star()
  update_Pf()
  update_k_for_pinning()
  update_kappa_avg()
```



Update flowing pigment

$$\mathbf{P_f}^* = sample(\mathbf{P_f}, \boldsymbol{x} - \boldsymbol{u})$$

Just use *taichi_glsl.sampling.bilerp()*

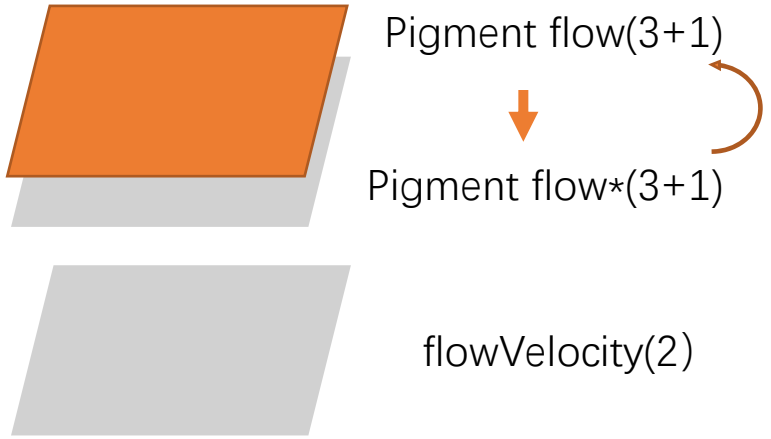


$$\mathbf{P_f} = mix(\mathbf{P_f}^*, \mathbf{P_f}, \gamma^*)$$

where

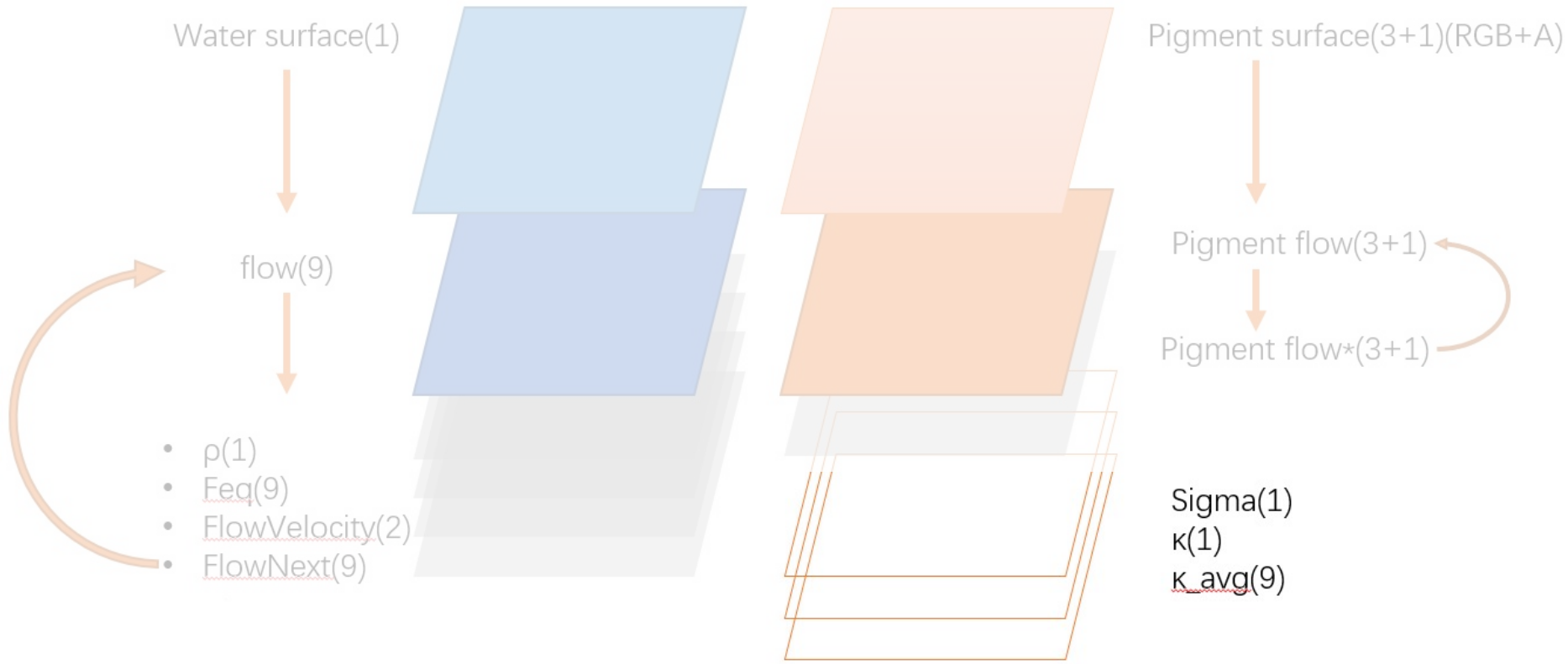
$$\gamma^* = mix(1.0, \gamma, smoothstep(|\boldsymbol{u}|, 0, \zeta))$$

γ ≈0.005, ζ ≈0.001



Overview

```
while Ture:
  drawing_on_surface()
  water_pigment_surface_to_flow()
  update_rho()
  update_Feq()
  update_Flow_Next()
  update_flow()
  update_Pf_star()
  update_Pf()
  update_k_for_pinning()
  update_kappa_avg()
```



Update **κ**

$$\sigma = q_1 + q_2 * fiber + q_3 * paper$$

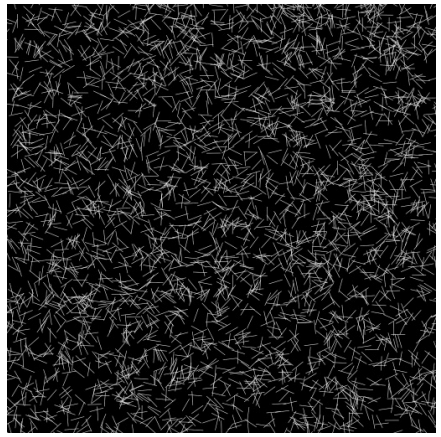
If a site P is dry and its neighbor’ s ρ are all under a threshold(given by sigma), then κ=0.99.

While drawing, the κ of the site P under brush become **Paper**[P].

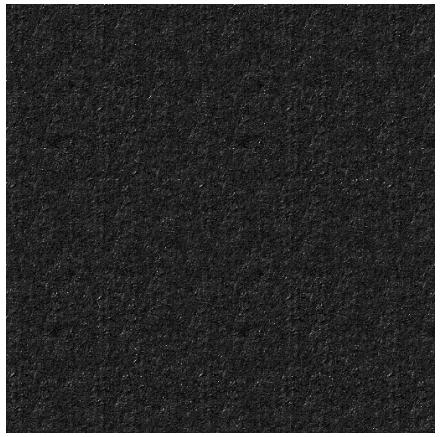
Then update $\bar{\boldsymbol{\kappa}}$.

$q_1 = 0.01$

$q_1 = -0.01$

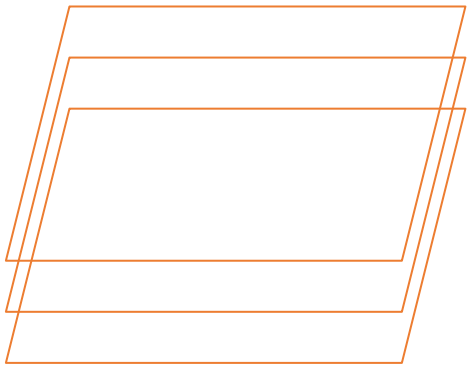


fiber



paper

$\sigma(1)$
$\kappa(1)$
$\bar{\boldsymbol{\kappa}}(9)$



Performance profiling

```
================================================================================
Kernel Profiler(count) @ CUDA on NVIDIA GeForce GTX 970
================================================================================
[      %     total   count |      min       avg       max   ] Kernel name
--------------------------------------------------------------------------------
[  6.95%   0.367 s   1000x |    0.013     0.367     1.646 ms] update_flow_c62_0_kernel_42_struct_for
[  6.33%   0.334 s   1000x |    0.294     0.334     1.427 ms] update_kappa_avg_c52_0_kernel_9_range_for
[  6.23%   0.329 s   1000x |    0.020     0.329     1.510 ms] update_FlowNext_c60_0_kernel_35_struct_for
[  4.38%   0.231 s   1000x |    0.015     0.231     1.411 ms] update_Feq_c64_0_kernel_28_struct_for
[  3.51%   0.185 s   1000x |    0.119     0.185     1.313 ms] render_c74_0_kernel_66_range_for
[  3.06%   0.162 s   1000x |    0.013     0.162     1.321 ms] update_Pf_star_c70_0_kernel_50_struct_for
[  2.84%   0.150 s   1000x |    0.015     0.150     1.303 ms] update_Pf_c72_0_kernel_57_struct_for
[  2.84%   0.150 s   1000x |    0.013     0.150     1.250 ms] update_rho_c58_0_kernel_20_struct_for
```

Thank you for listening

Github: https://github.com/Vineyo/Taichi-Moxi

Taichi forum: https://forum.taichi.graphics/t/topic/1810