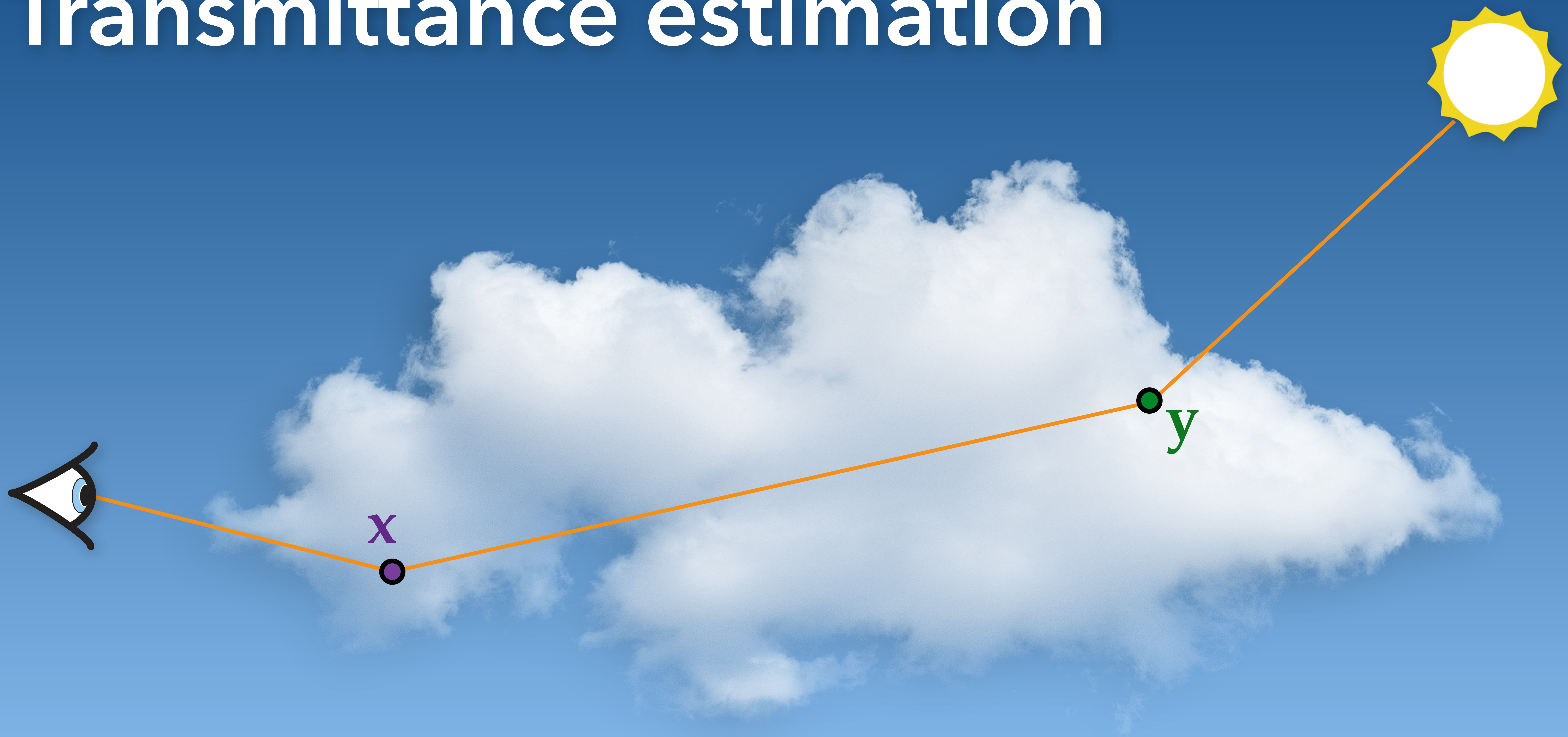
Monte Carlo Methods for Physically Based Volume Rendering

Transmittance estimation

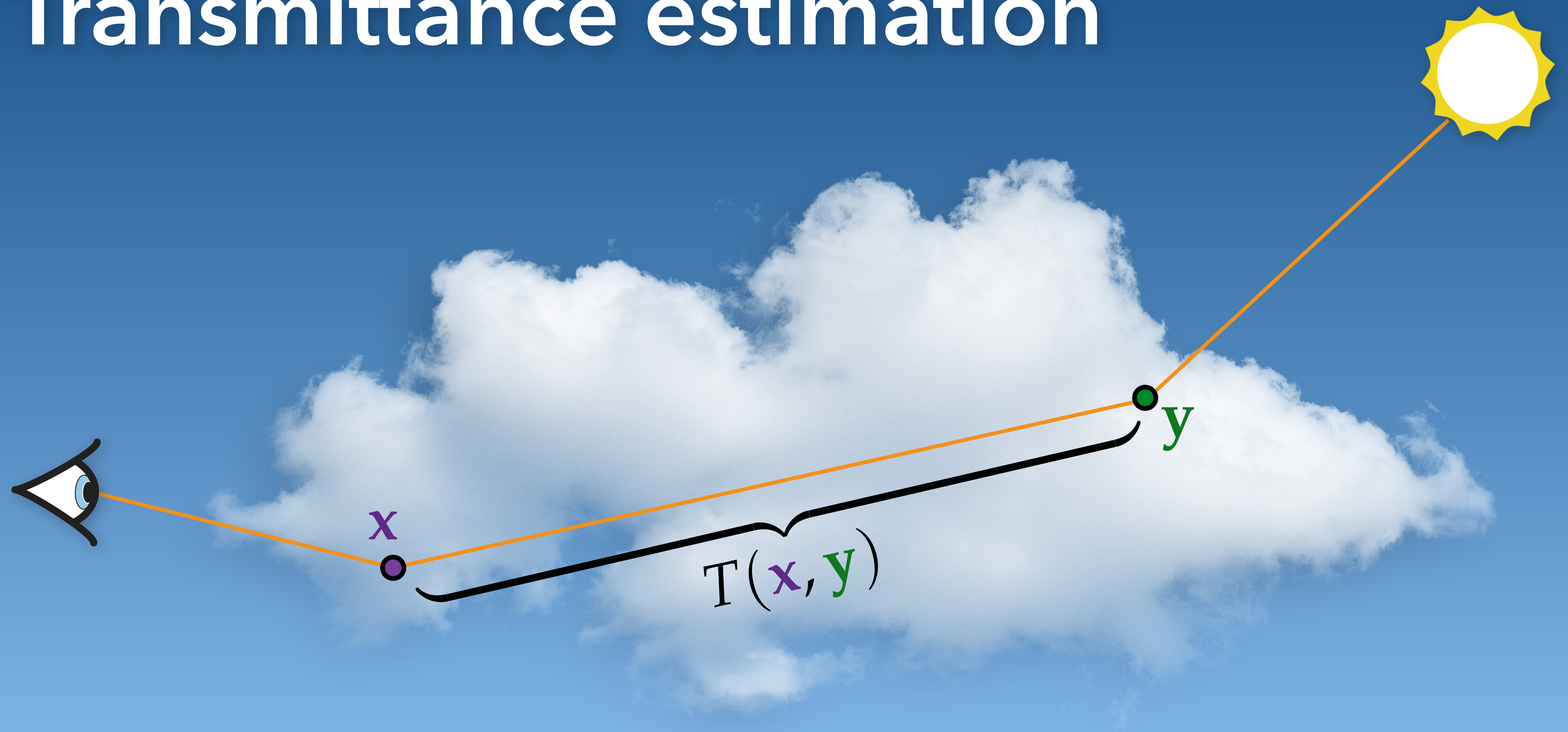
Wojciech Jarosz
wjarosz@dartmouth.edu

DARTMOUTH
VISUAL COMPUTING LAB

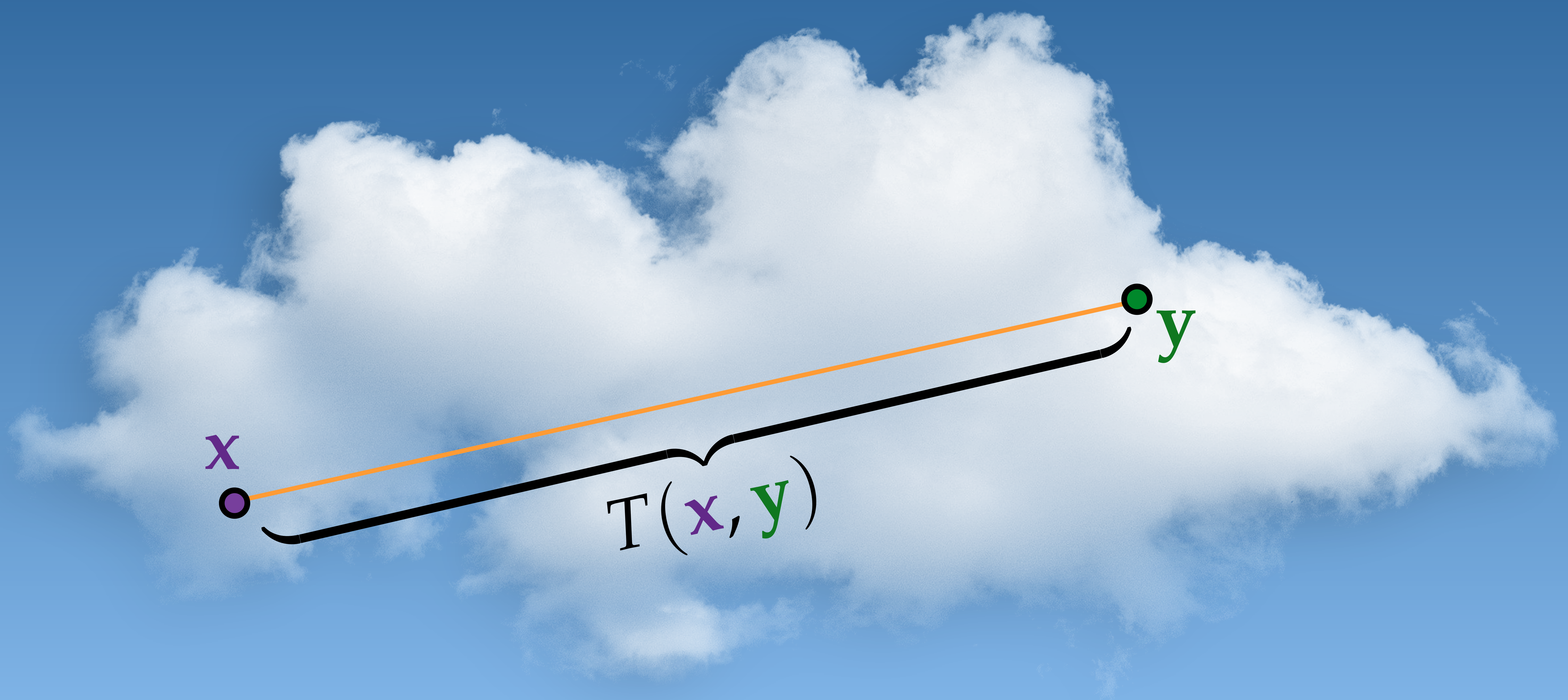
Transmittance estimation



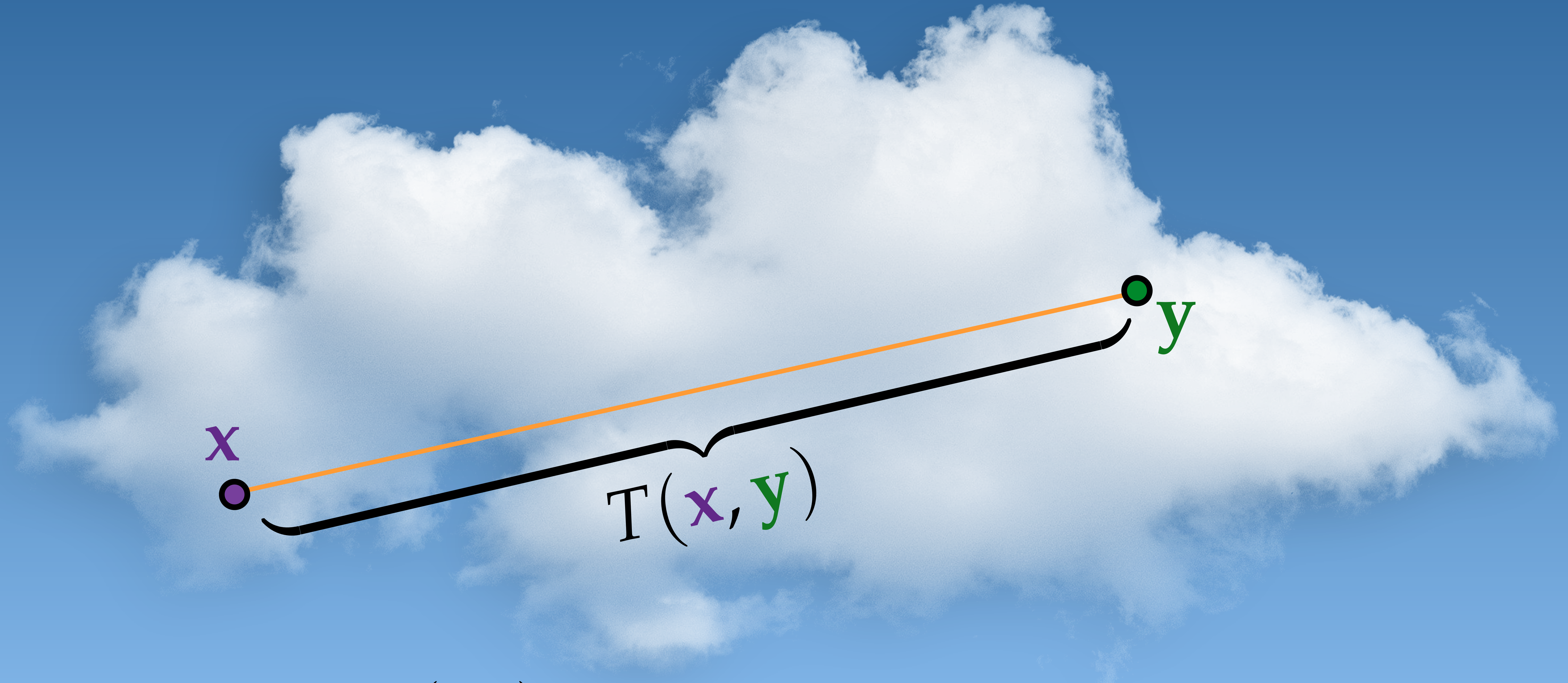
Transmittance estimation



Transmittance estimation



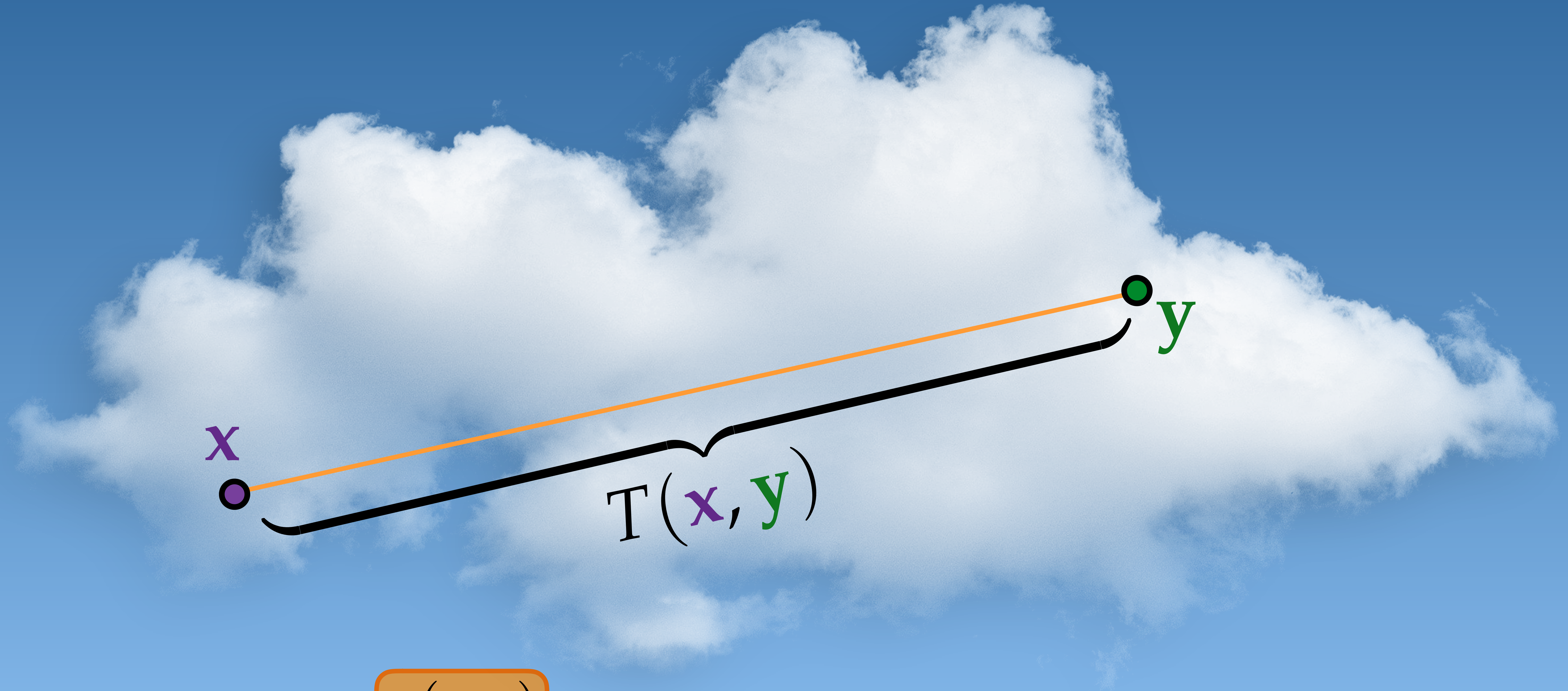
Transmittance estimation



$$T(\mathbf{x}, \mathbf{y}) = e^{-\tau(\mathbf{x}, \mathbf{y})}$$

transmittance

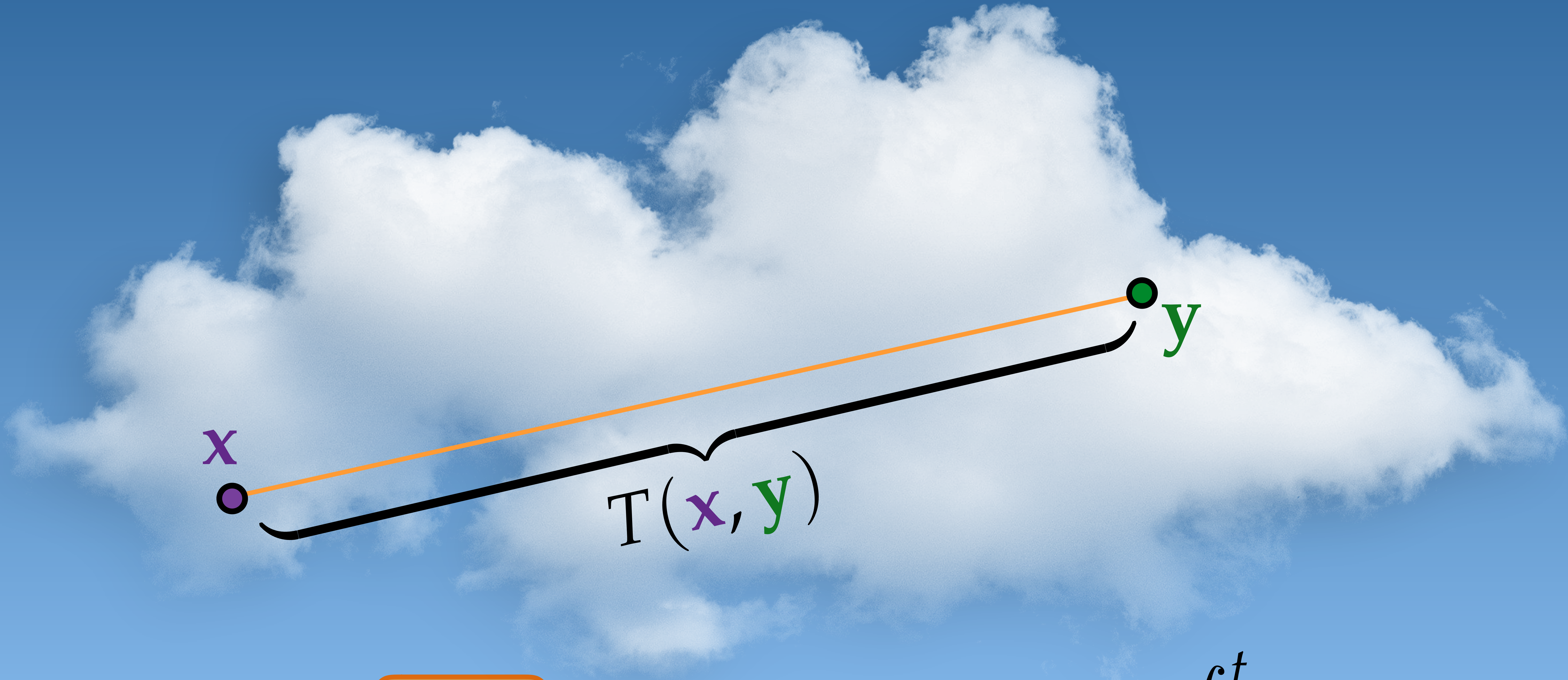
Transmittance estimation



$$T(\mathbf{x},\mathbf{y}) = e^{-\tau(\mathbf{x},\mathbf{y})}$$

transmittance

Transmittance estimation



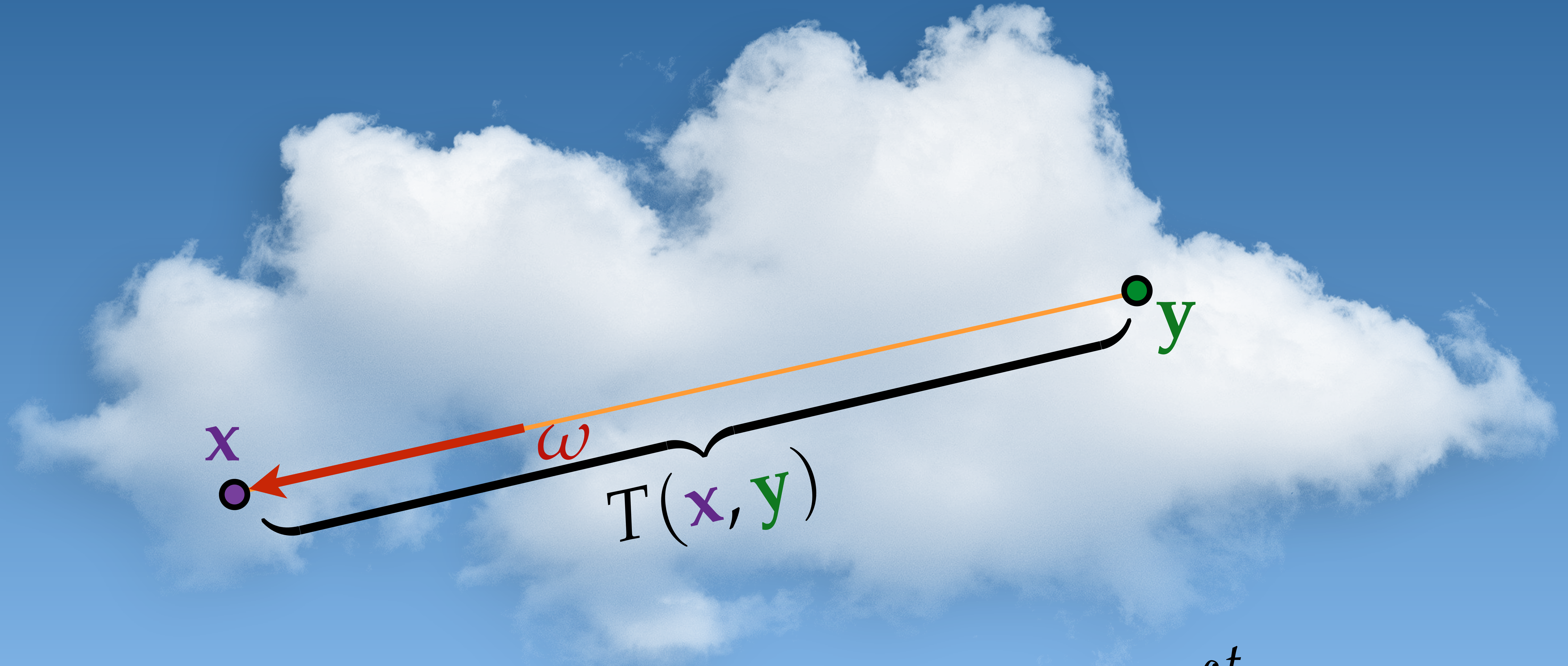
$$T(\mathbf{x}, \mathbf{y}) = e^{-\tau(\mathbf{x}, \mathbf{y})}$$

transmittance

$$\tau(\mathbf{x}, \mathbf{y}) = \int_0^t \mu_t(\mathbf{x} - s\omega)\, ds$$

"optical thickness"

Transmittance estimation



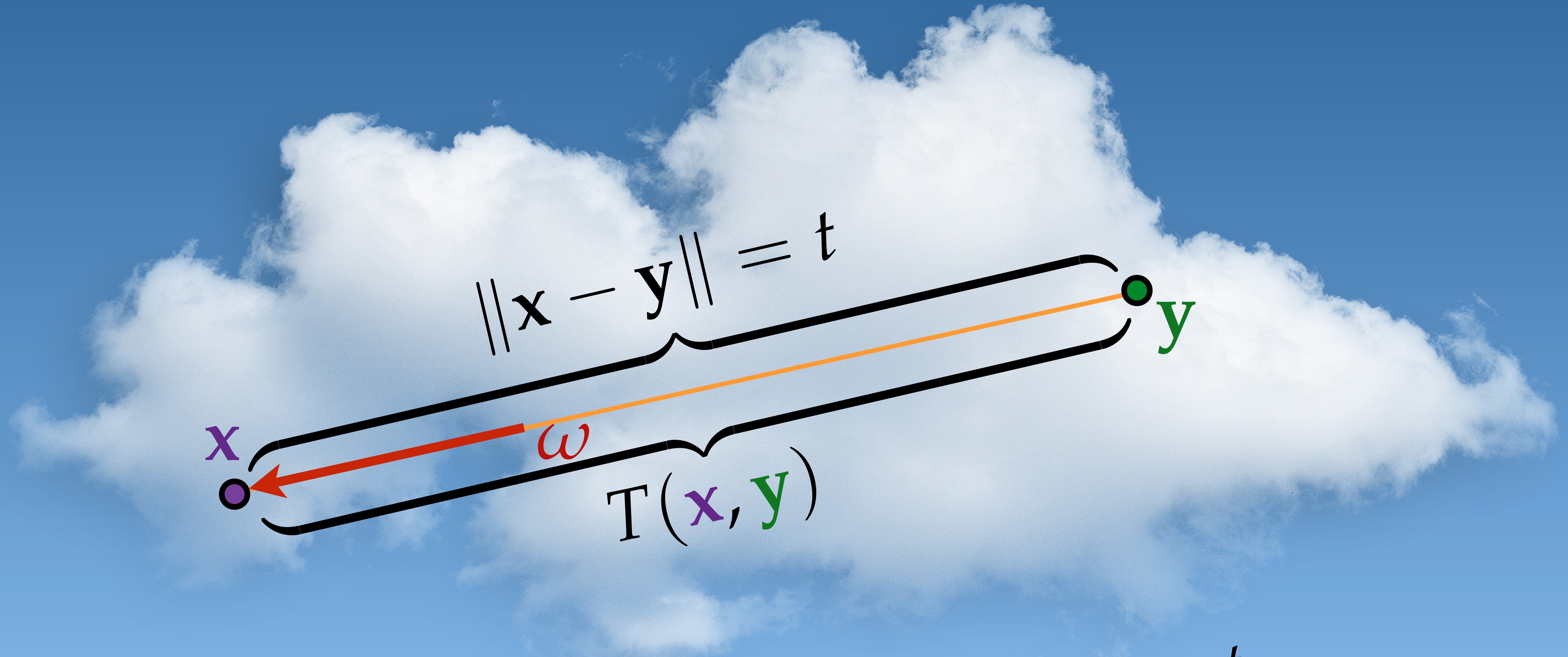
$$T(\mathbf{x},\mathbf{y}) = e^{-\tau(\mathbf{x},\mathbf{y})}$$

transmittance

$$\tau(\mathbf{x},\mathbf{y}) = \int_0^t \mu_t(\mathbf{x} - s\omega)\, ds$$

"optical thickness"

Transmittance estimation



$$T(\mathbf{x},\mathbf{y}) = e^{-\tau(\mathbf{x},\mathbf{y})}$$

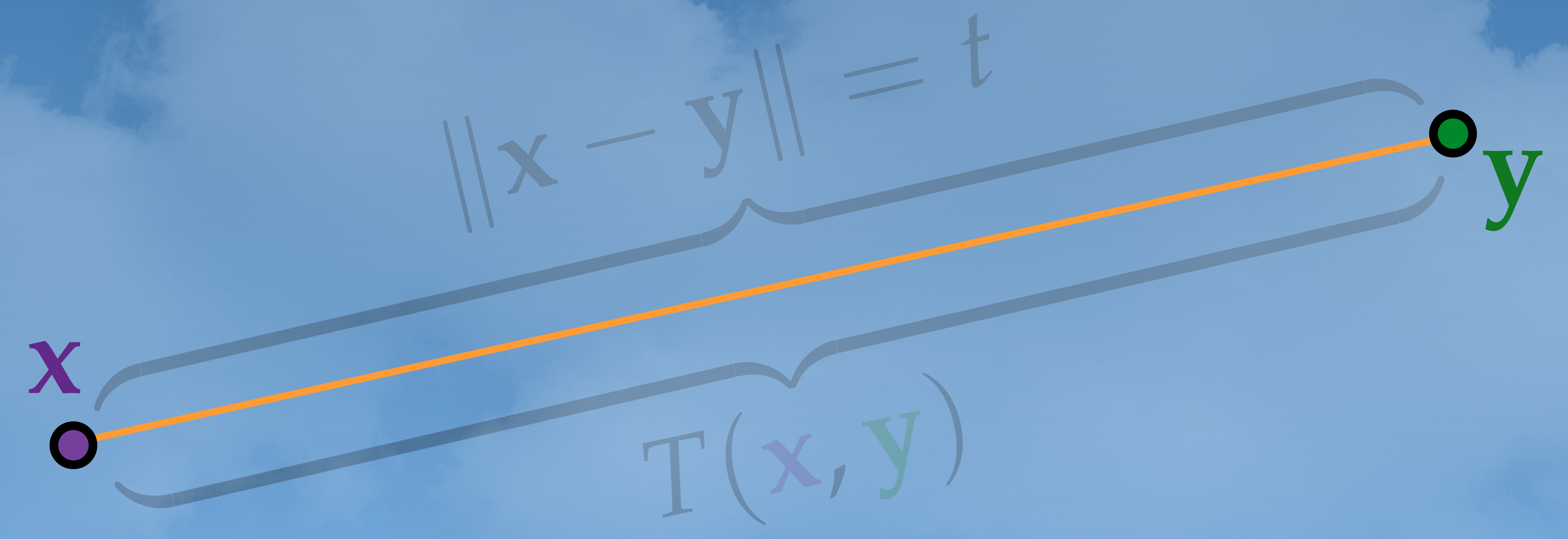
transmittance

$$\tau(\mathbf{x},\mathbf{y}) = \int_0^t \mu_t(\mathbf{x} - s\omega)\,\mathrm{d}s$$

"optical thickness"

Transmittance estimation

1. Estimators integrating **optical thickness**



$$T(\mathbf{x}, \mathbf{y}) = \mathrm{e}^{-\tau(\mathbf{x}, \mathbf{y})}$$

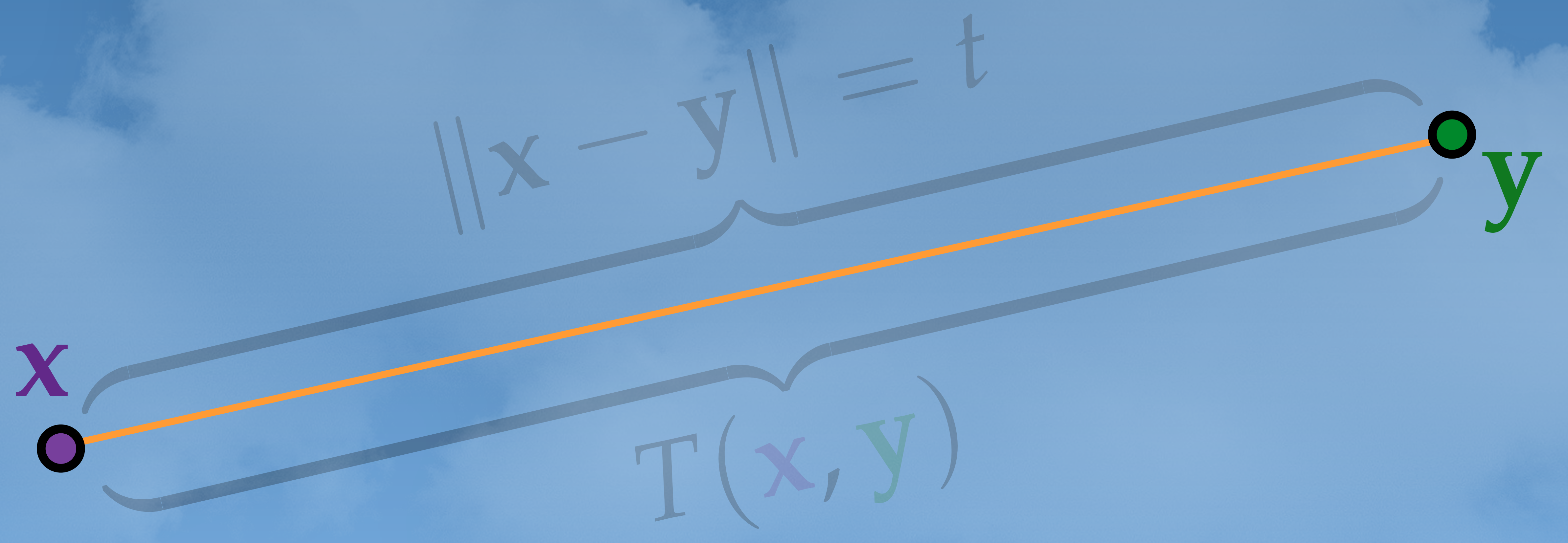
transmittance

$$\tau(\mathbf{x}, \mathbf{y}) = \int_0^t \mu_{\mathrm{t}}(\mathbf{x} - s\omega)\, \mathrm{d}s$$

"optical thickness"

Transmittance estimation

1. Estimators integrating **optical thickness**
2. Estimators using **free-flight sampling**



$$T(\mathbf{x},\mathbf{y}) = \mathrm{e}^{-\tau(\mathbf{x},\mathbf{y})}$$

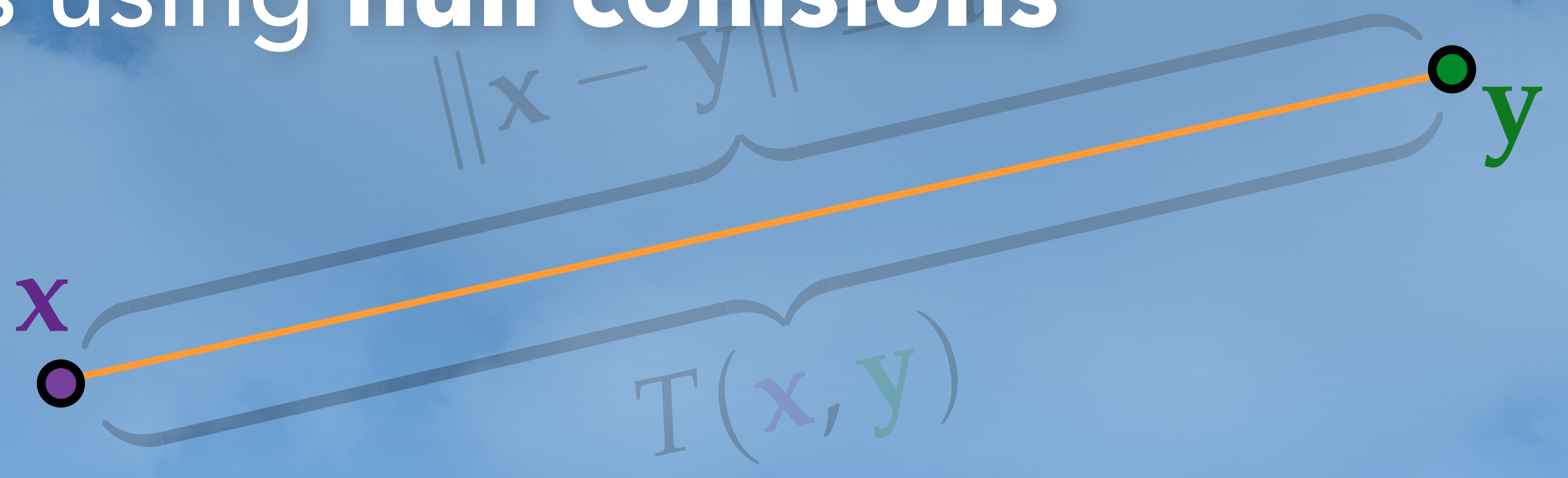
transmittance

$$\tau(\mathbf{x},\mathbf{y}) = \int_0^t \mu_\mathrm{t}(\mathbf{x} - s\omega)\,\mathrm{d}s$$

"optical thickness"

Transmittance estimation

1. Estimators integrating **optical thickness**
2. Estimators using **free-flight sampling**
3. Estimators using **null collisions**



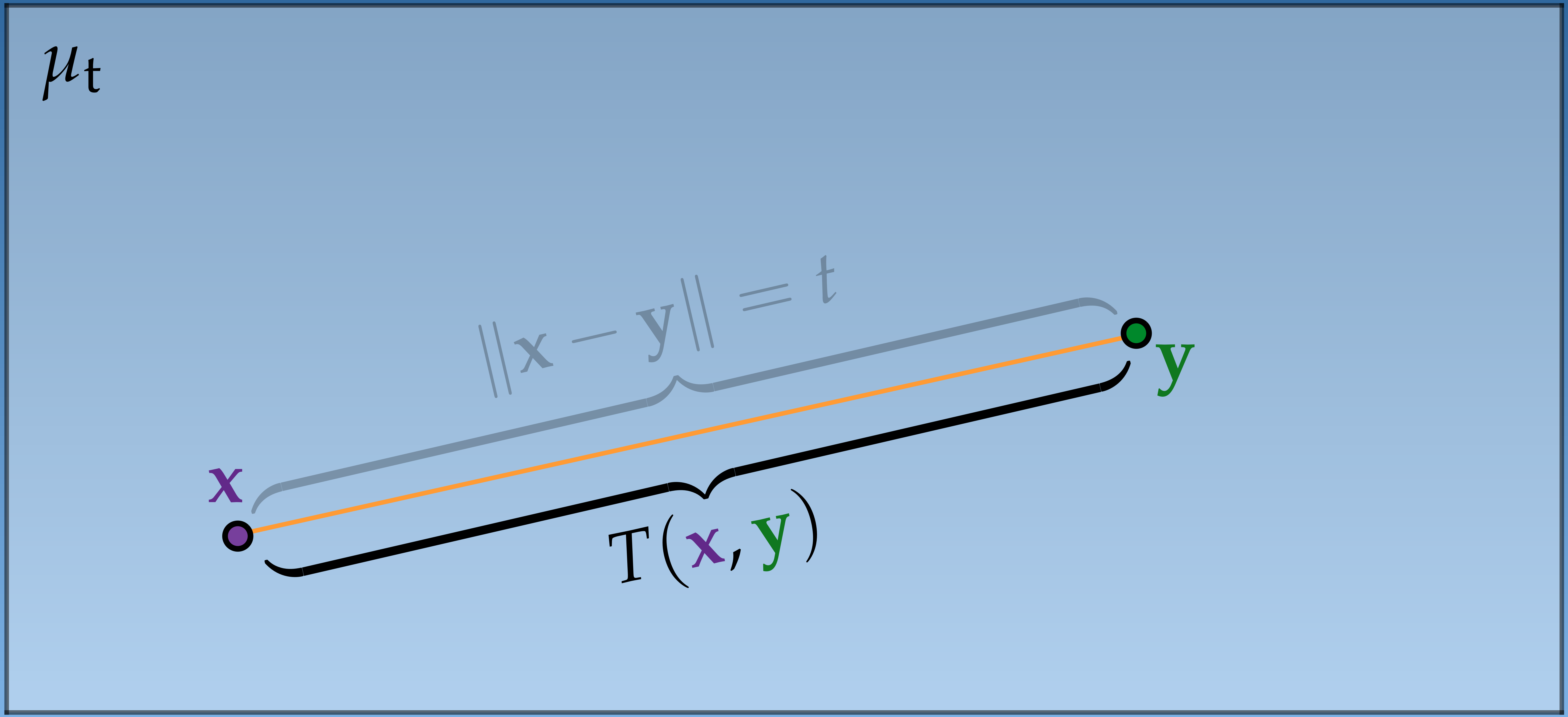
$$T(\mathbf{x}, \mathbf{y}) = \mathrm{e}^{-\tau(\mathbf{x}, \mathbf{y})}$$

transmittance

$$\tau(\mathbf{x}, \mathbf{y}) = \int_0^t \mu_\mathrm{t}(\mathbf{x} - s\omega)\,\mathrm{d}s$$

"optical thickness"

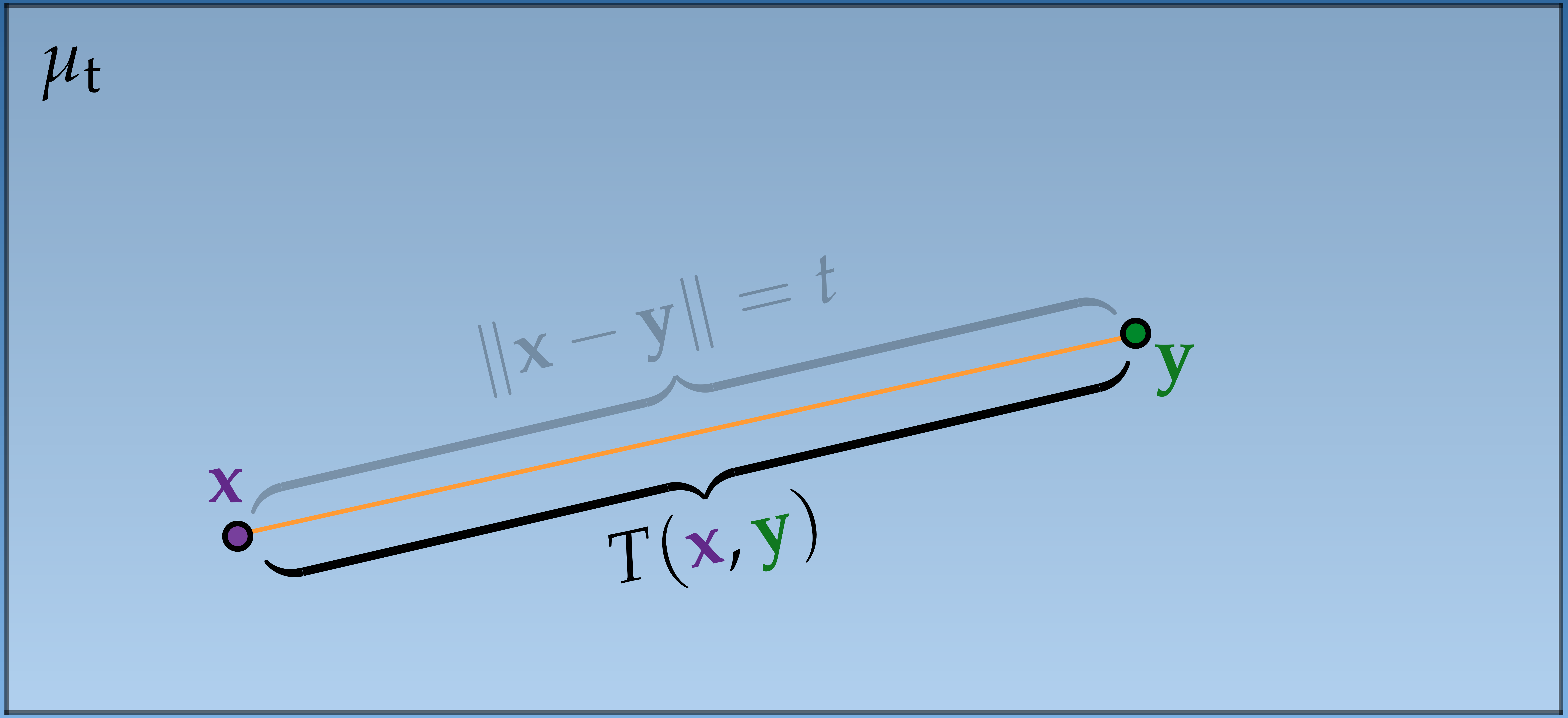
Homogeneous medium



$$T(\mathbf{x},\mathbf{y}) = \mathrm{e}^{-\tau(\mathbf{x},\mathbf{y})}$$

$$\tau(\mathbf{x},\mathbf{y}) = \int_0^t \mu_t(\mathbf{x} - s\omega)\,\mathrm{d}s$$

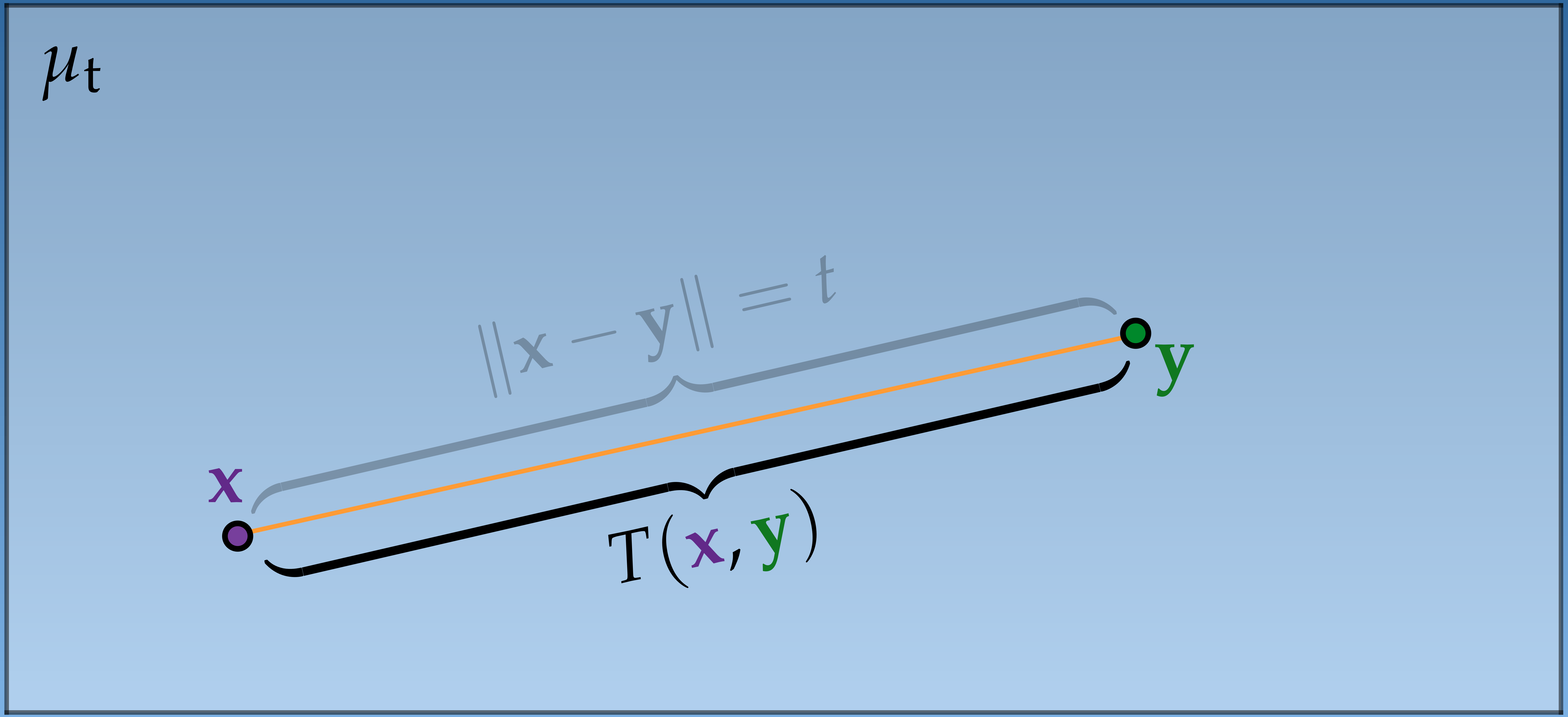
Homogeneous medium



$$T(\mathbf{x},\mathbf{y}) = e^{-\tau(\mathbf{x},\mathbf{y})}$$

$$\tau(\mathbf{x},\mathbf{y}) = \int_0^t \mu_t(\mathbf{x}-s\omega)\,ds = \mu_t\, t$$

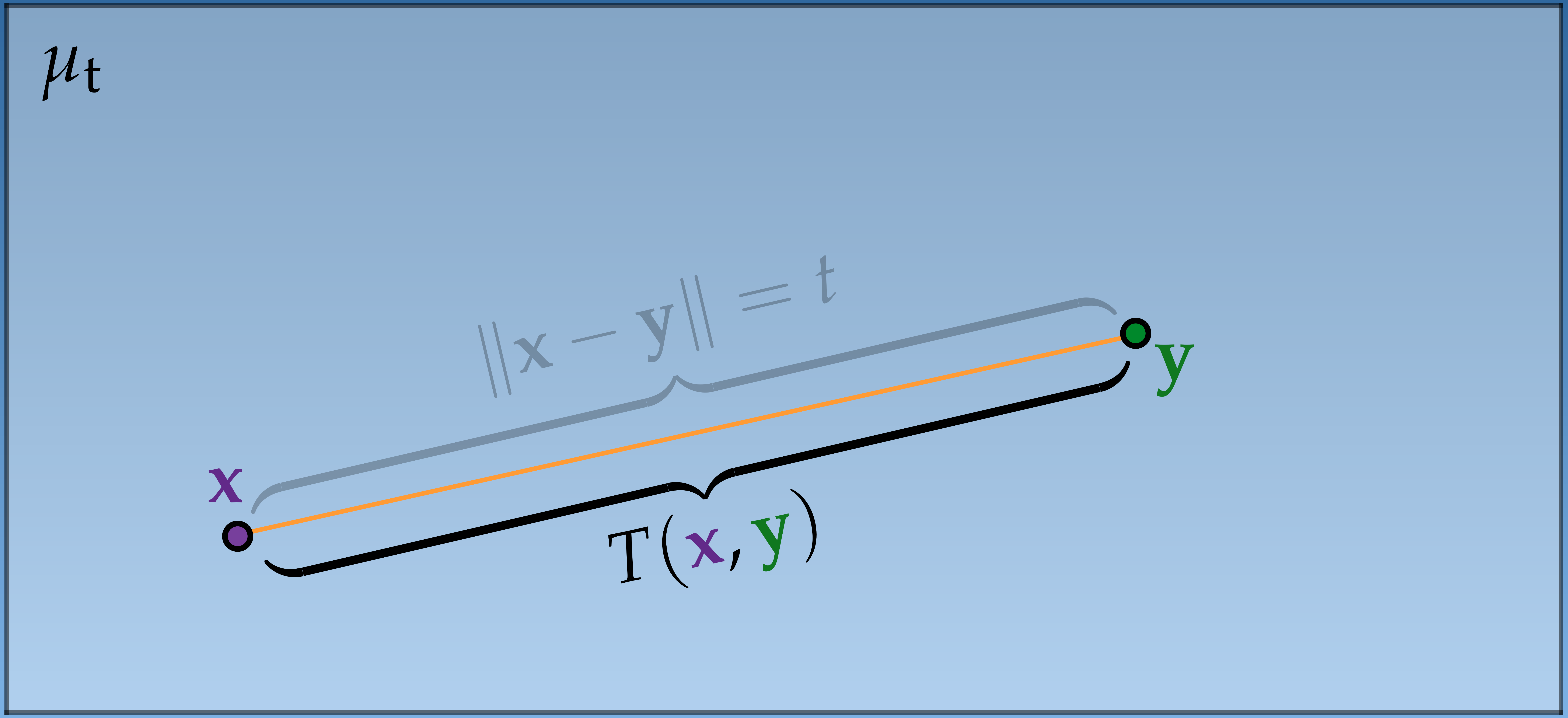
Homogeneous medium



$T(\mathbf{x}, \mathbf{y}) = e^{-}$

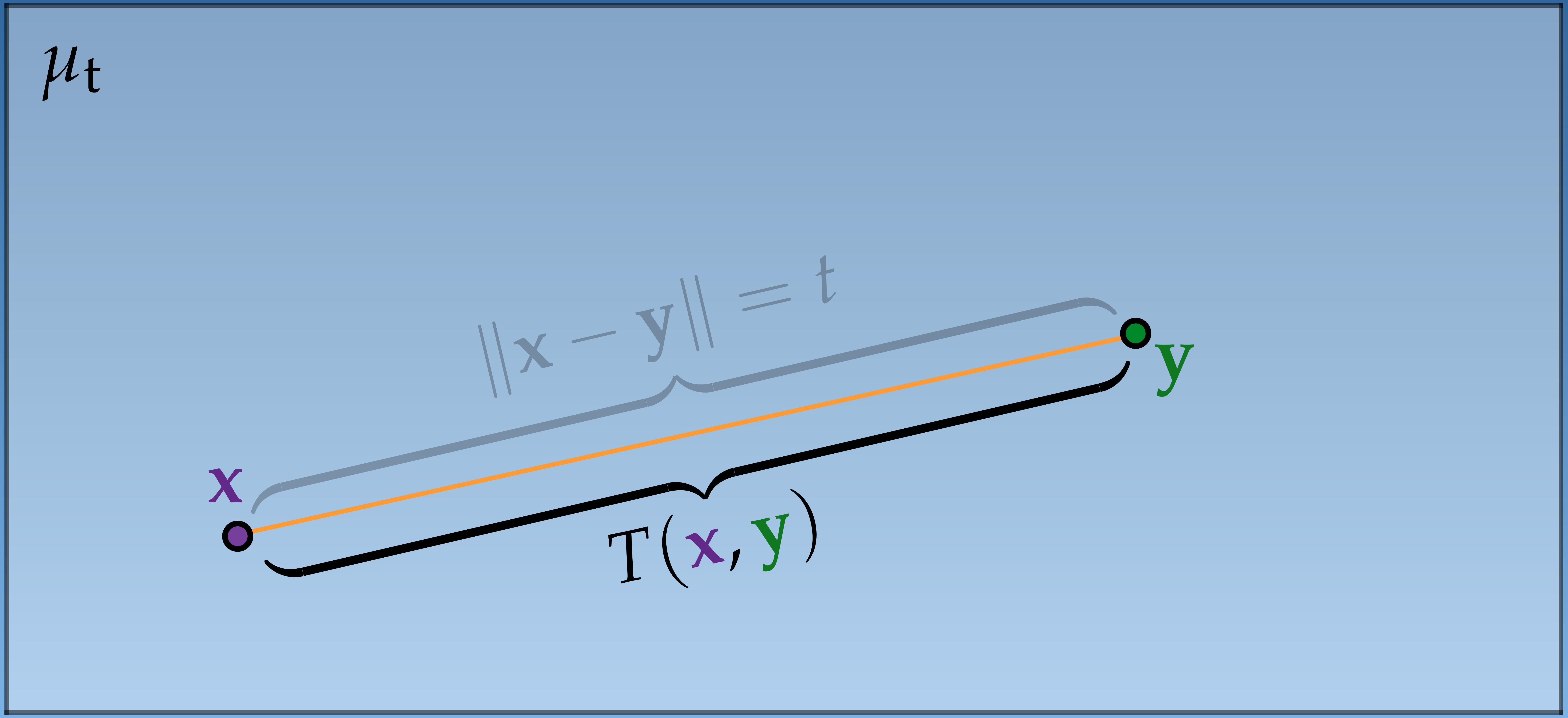
$\mu_t \, t$

Homogeneous medium



$$T(\mathbf{x},\mathbf{y}) = e^{-\mu_t t}$$

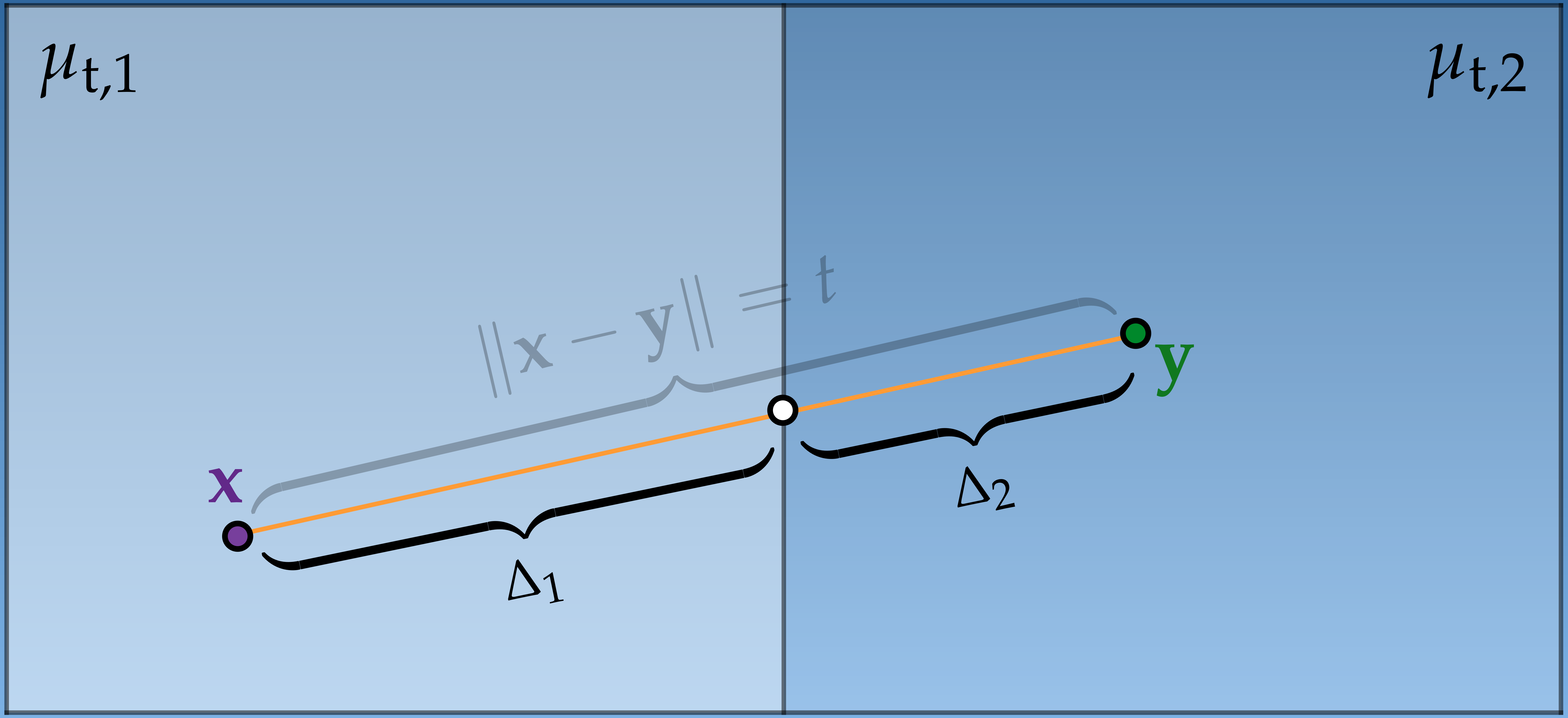
Homogeneous medium



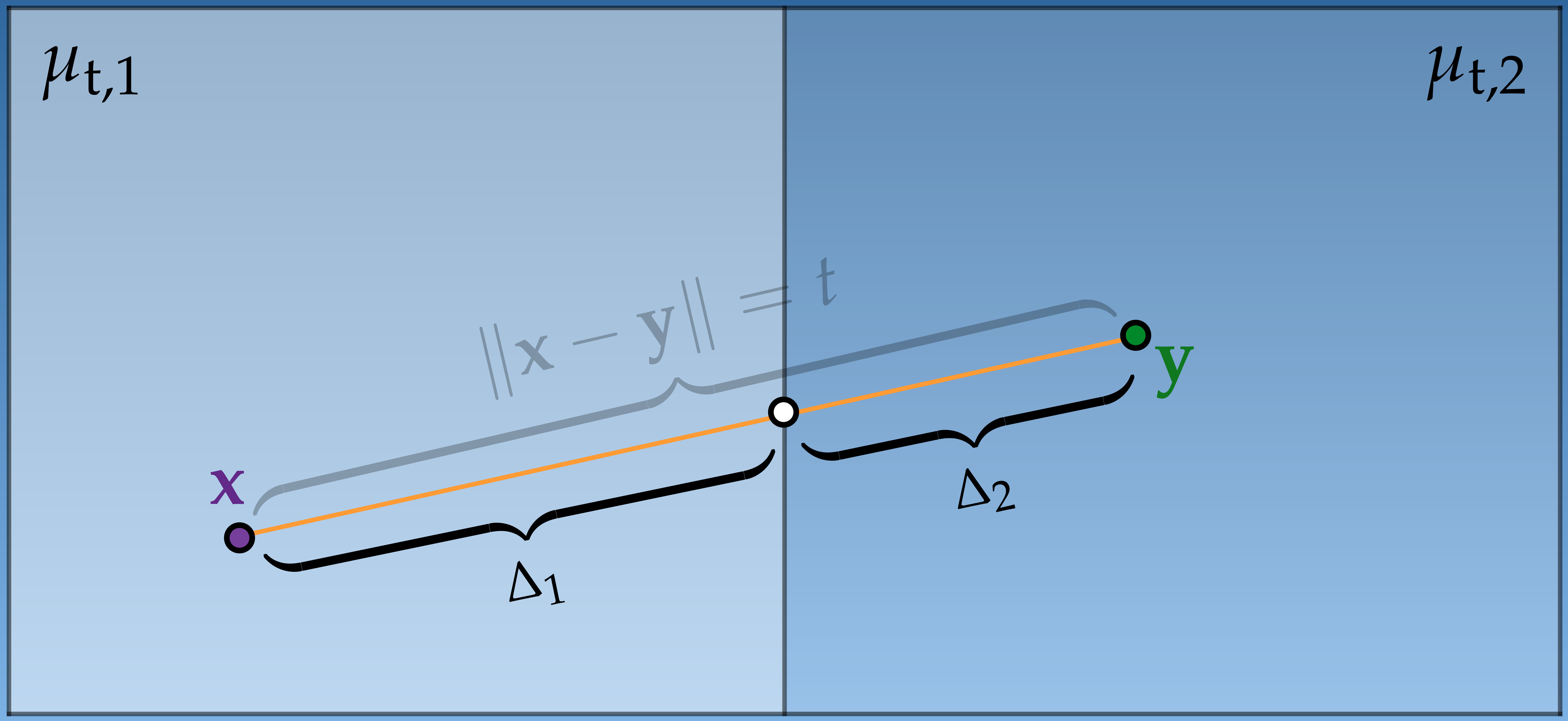
$$T(\mathbf{x}, \mathbf{y}) = e^{-\mu_t t}$$

$\langle T(t) \rangle_{EV}$: "**E**xpected **V**alue" estimator

Piecewise homogeneous medium

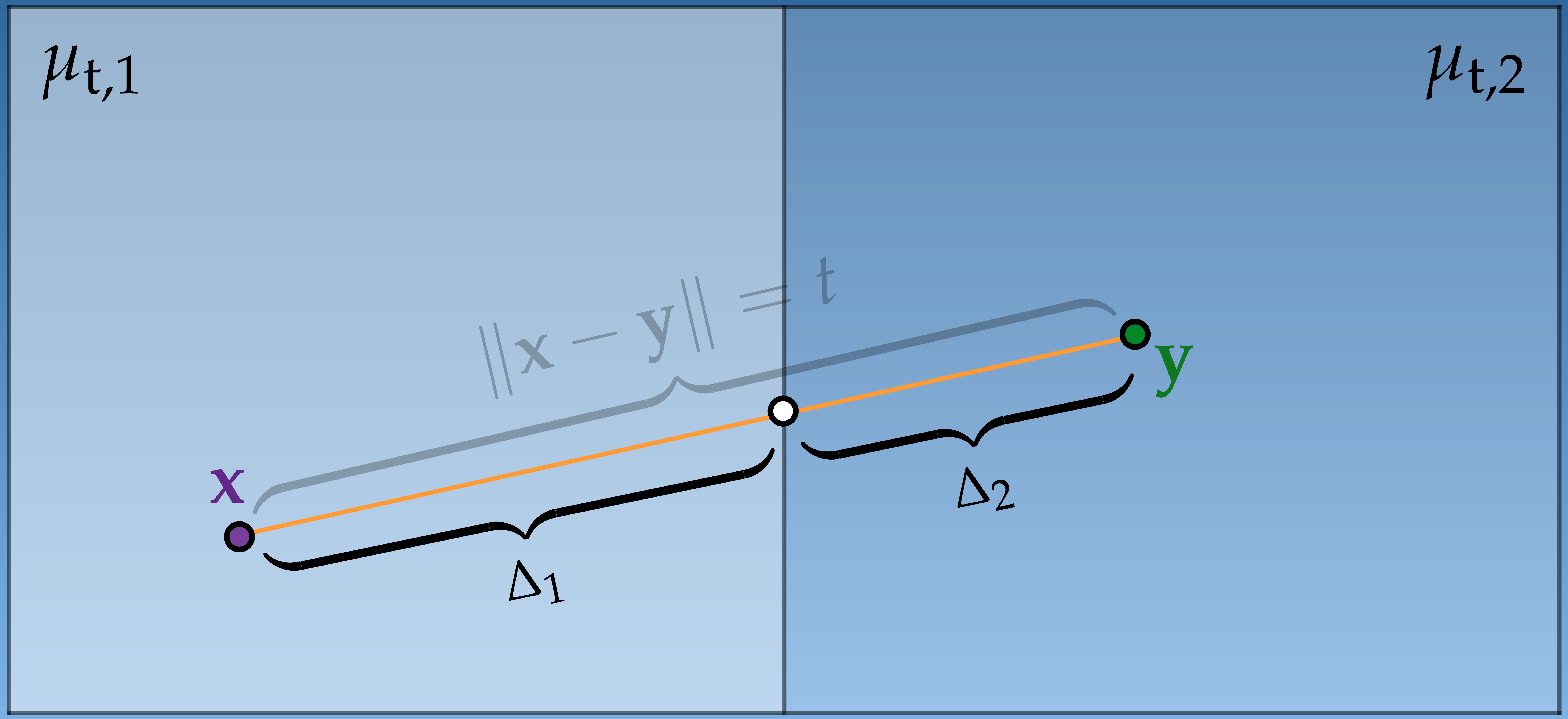


Piecewise homogeneous medium



$$t = \Delta_1 + \Delta_2$$

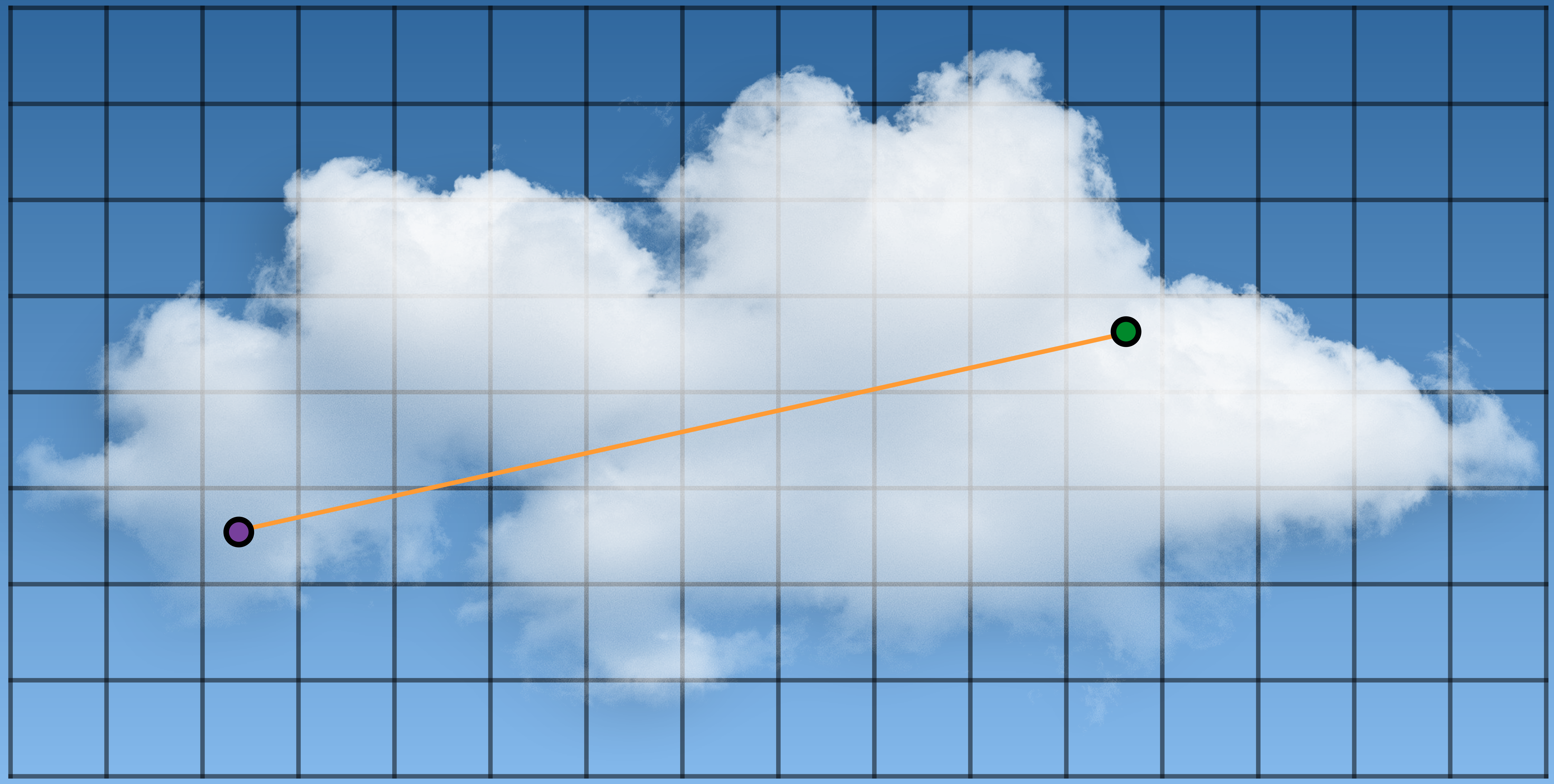
Piecewise homogeneous medium



$$t = \Delta_1 + \Delta_2$$

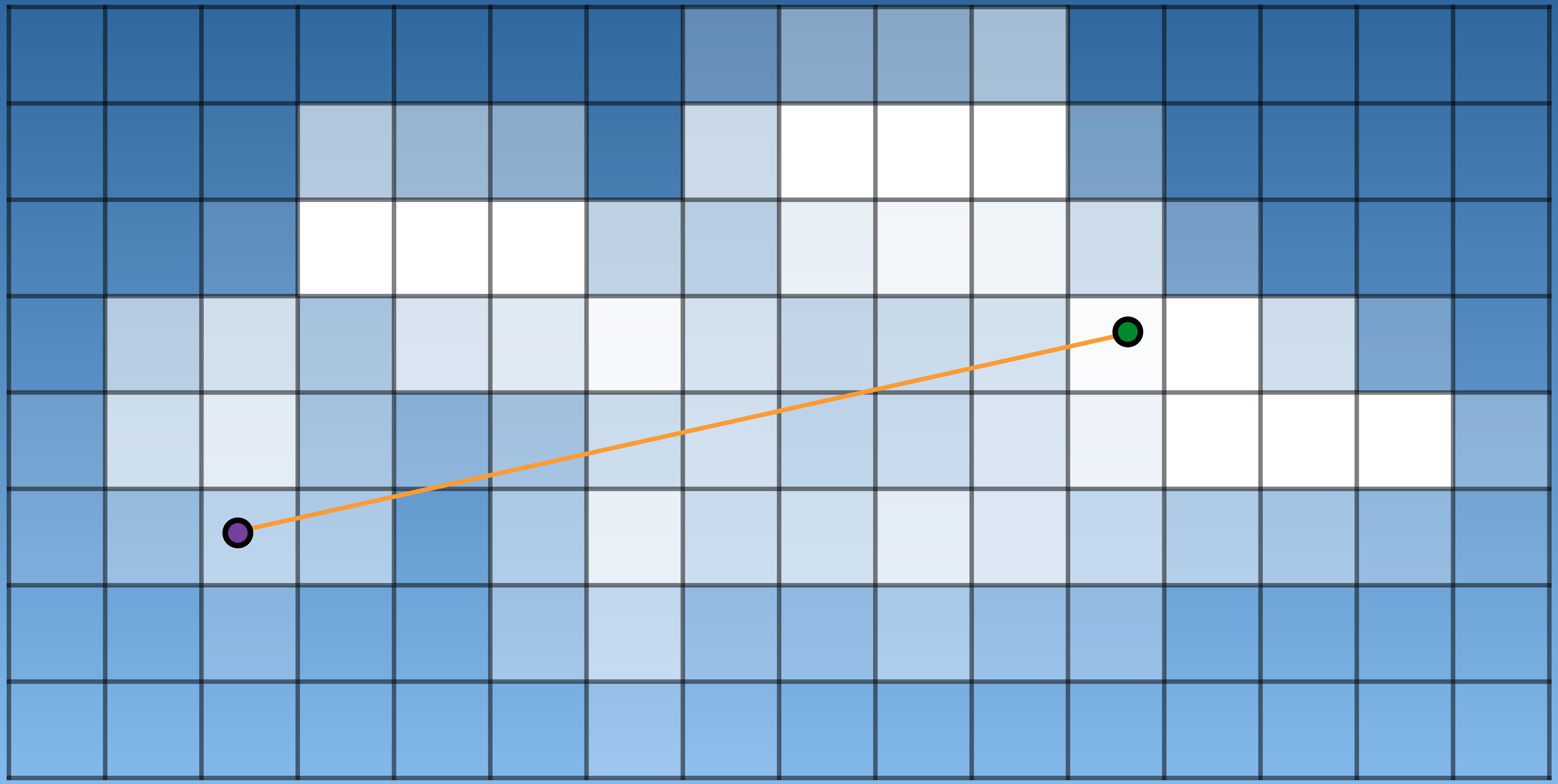
$$T(t) = e^{-\tau(t)} = e^{-(\tau_1+\tau_2)} = e^{-(\mu_{t,1}\Delta_1+\mu_{t,2}\Delta_2)}$$

Voxelize medium



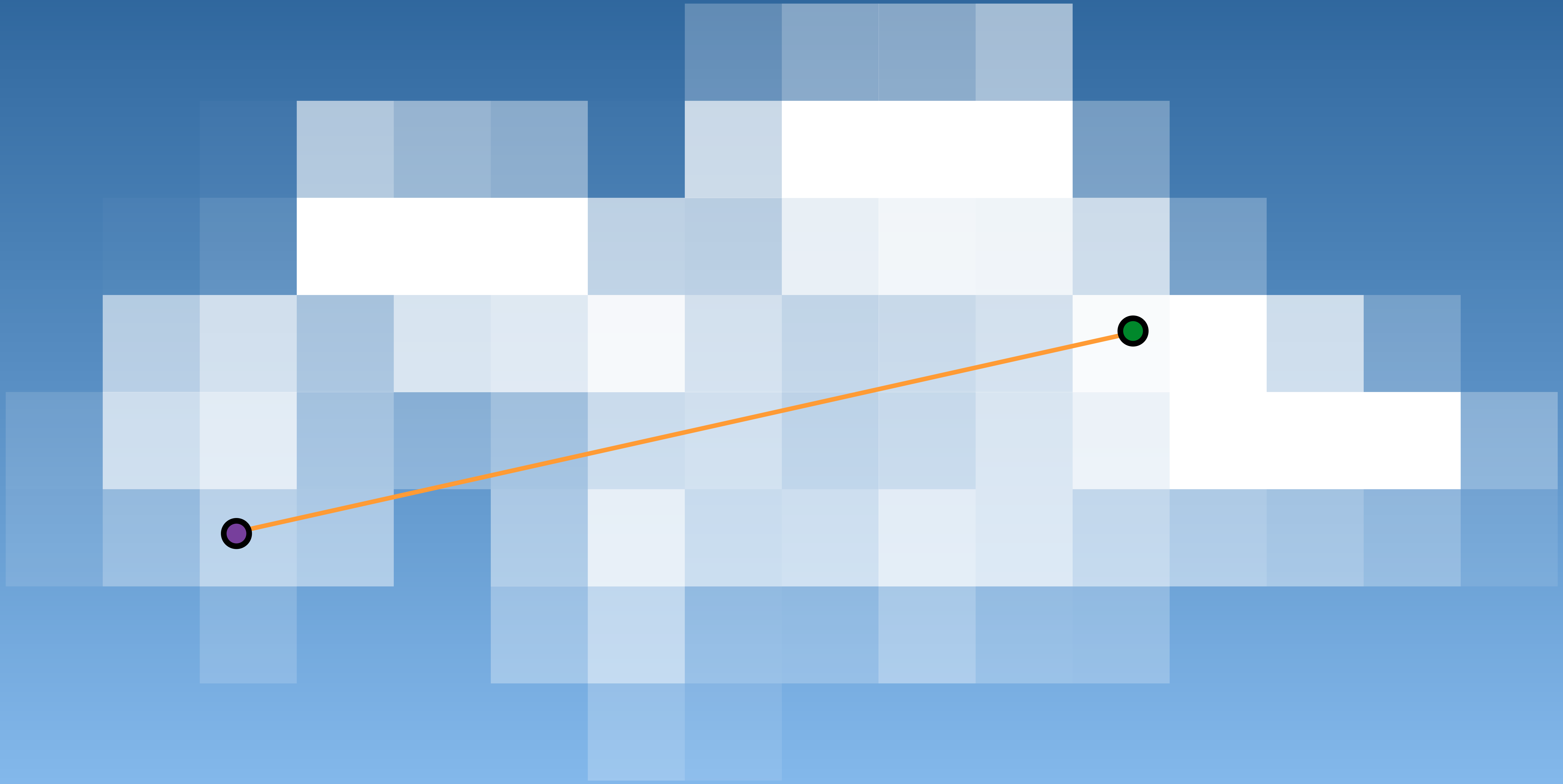
$$T(t) = \mathrm{e}^{-\tau(t)} = \mathrm{e}^{-(\tau_1+\tau_2)} = \mathrm{e}^{-(\mu_{\mathrm{t},1}\Delta_1+\mu_{\mathrm{t},2}\Delta_2)}$$

Voxelized medium (piecewise const.)



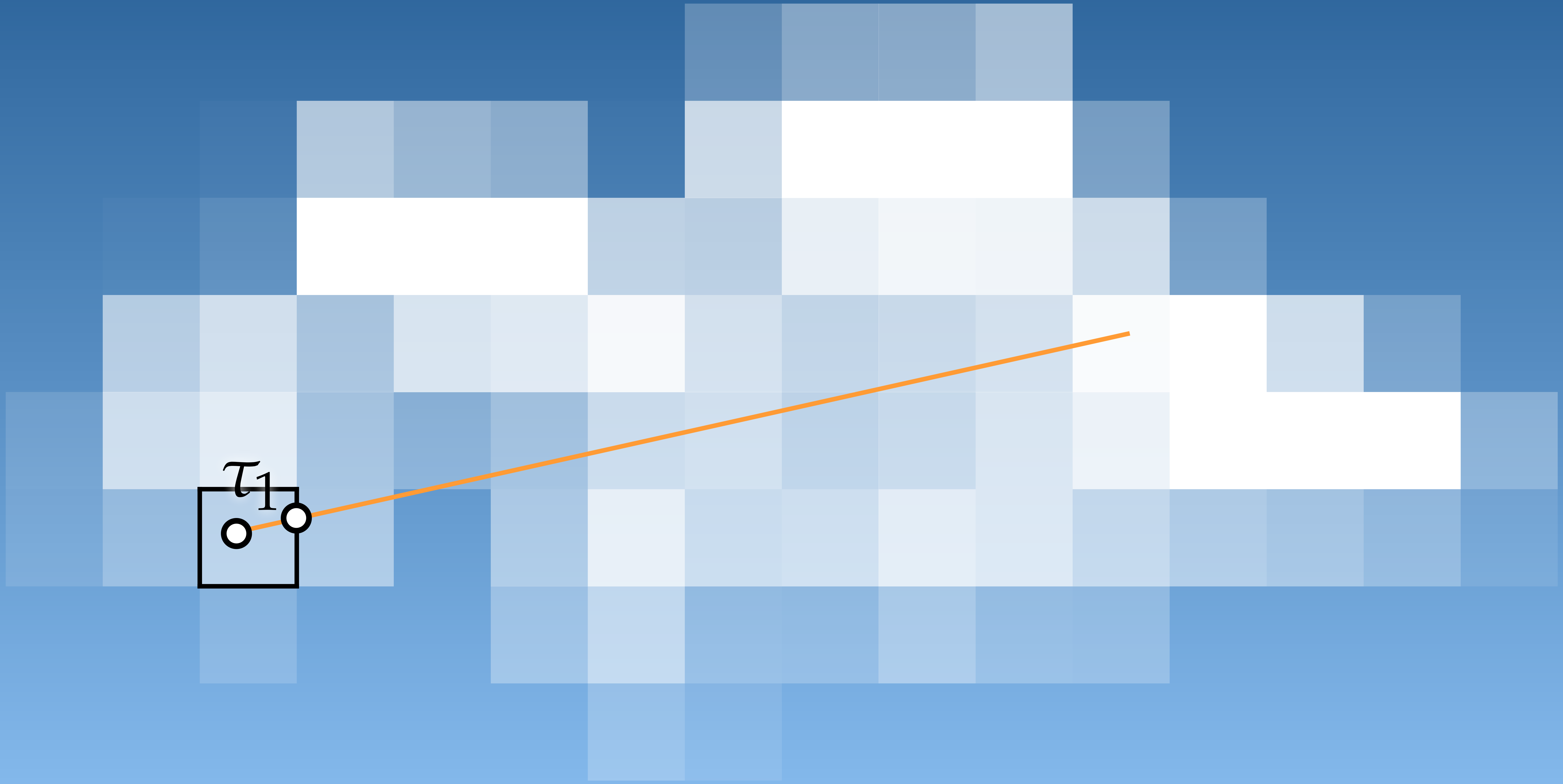
$$T(t) = \mathrm{e}^{-\tau(t)} = \mathrm{e}^{-\sum_i^k \tau_i} = \mathrm{e}^{-\sum_i^k \mu_{\mathrm{t},i}\,\Delta_i}$$

Voxelized medium (piecewise const.)



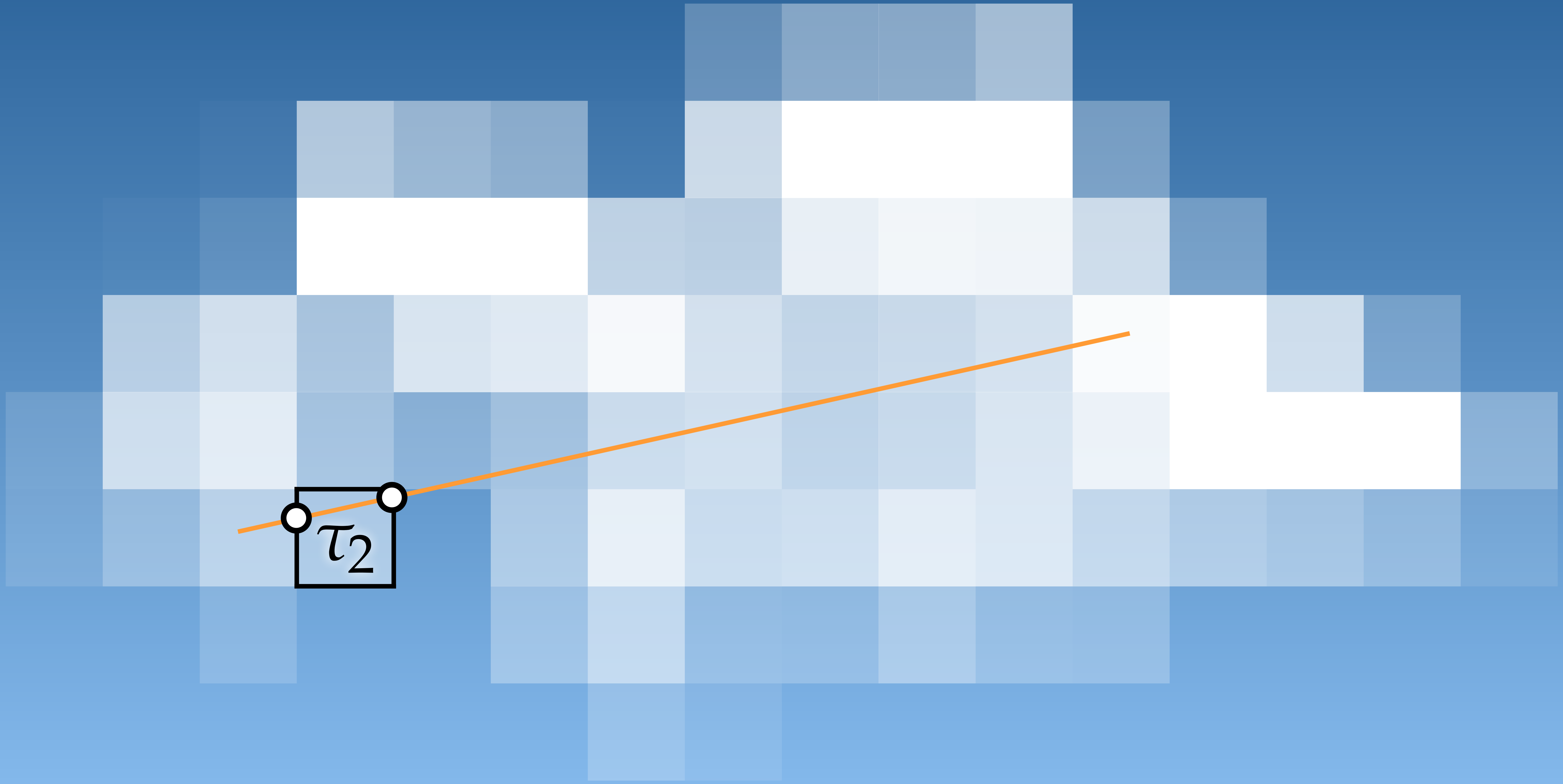
$$T(t) = e^{-\tau(t)} = e^{-\sum_i^k \tau_i} = e^{-\sum_i^k \mu_{t,i}\,\Delta_i}$$

Regular tracking (piecewise constant)



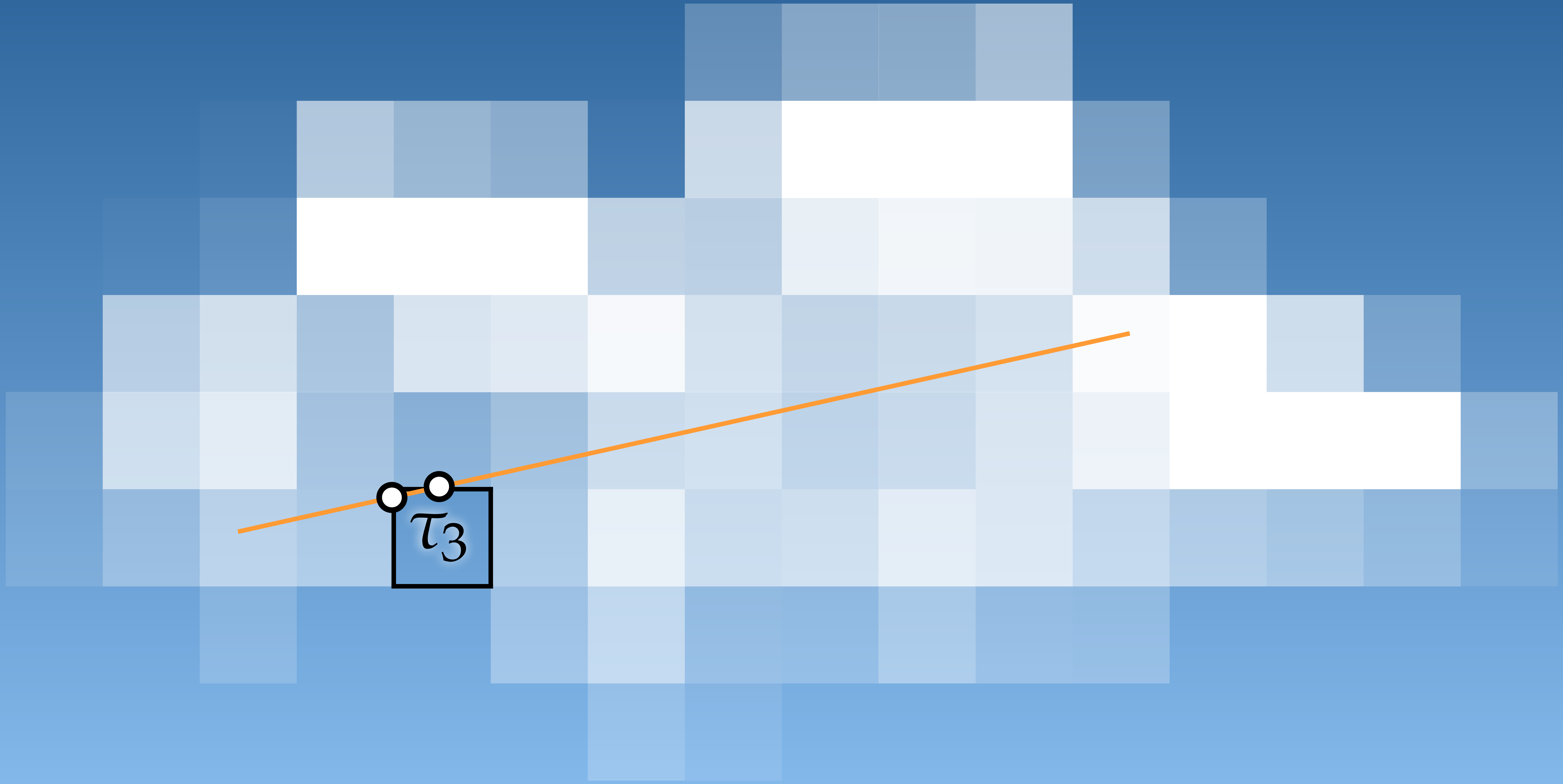
$$T(t) = \mathrm{e}^{-\tau(t)} = \mathrm{e}^{-\sum_i^k \tau_i} = \mathrm{e}^{-\sum_i^k \mu_{\mathrm{t},i}\,\Delta_i}$$

Regular tracking (piecewise constant)



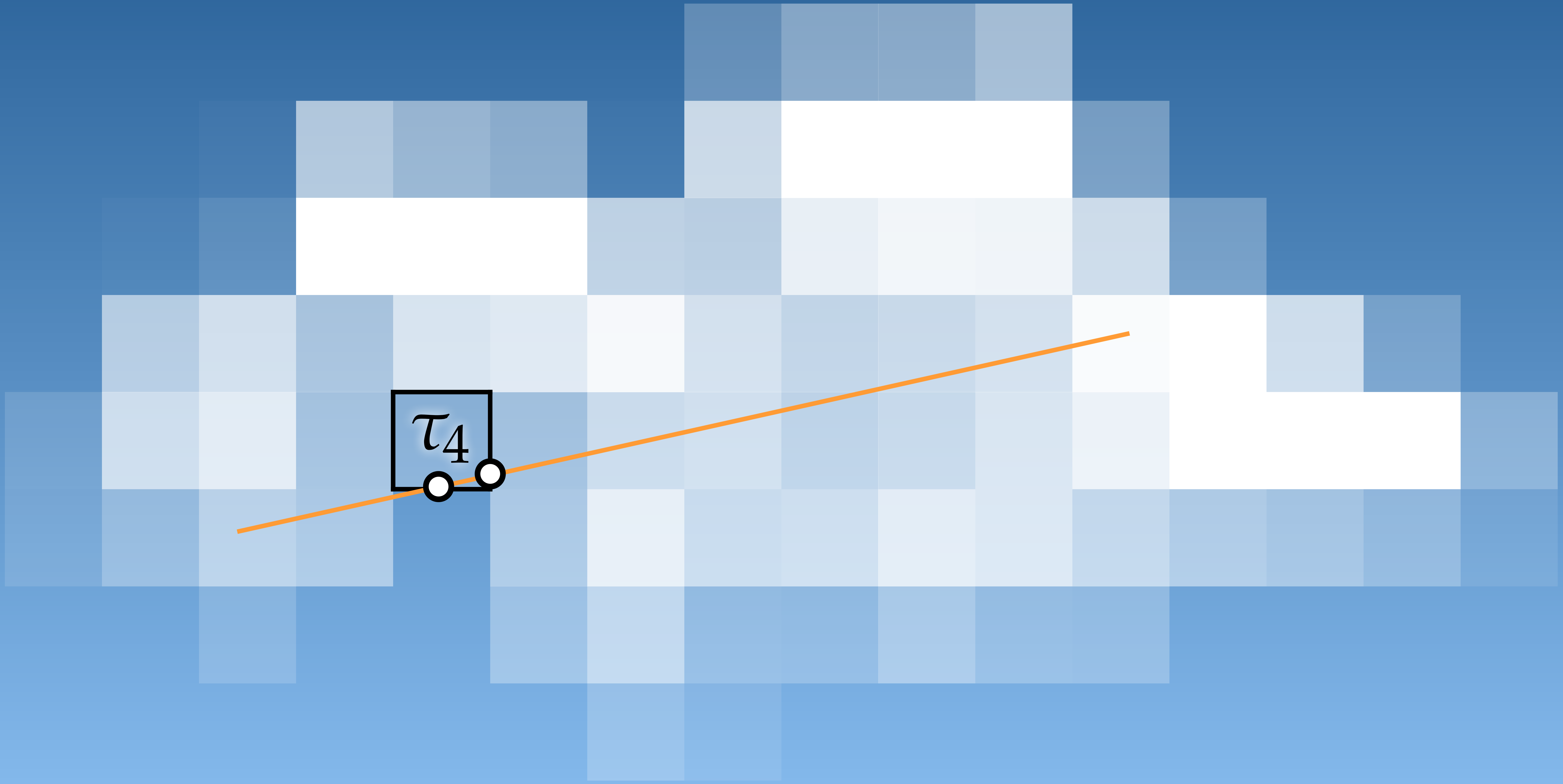
$$T(t) = \mathrm{e}^{-\tau(t)} = \mathrm{e}^{-\sum_i^k \tau_i} = \mathrm{e}^{-\sum_i^k \mu_{\mathrm{t},i}\,\Delta_i}$$

Regular tracking (piecewise constant)



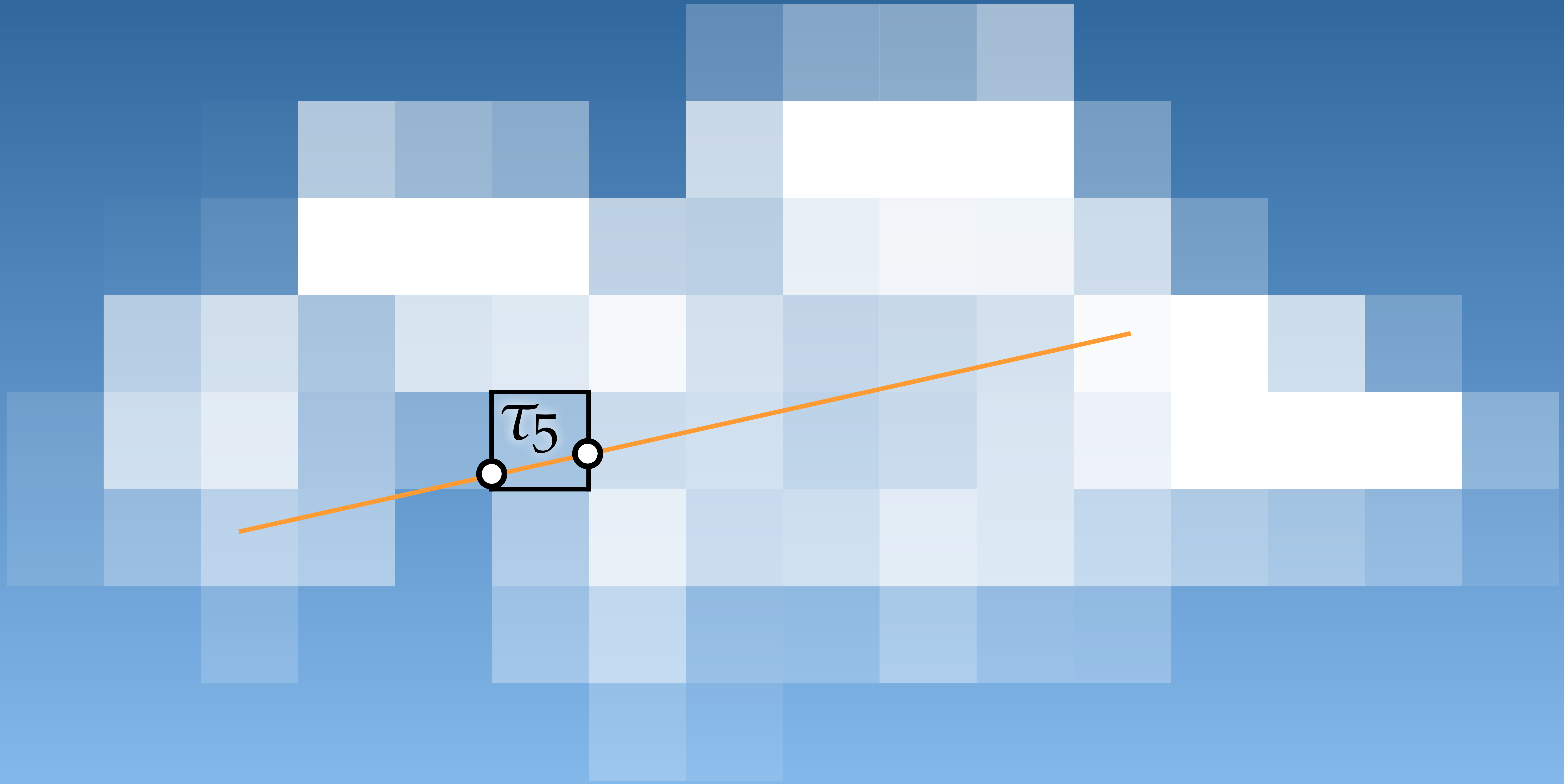
$$T(t) = e^{-\tau(t)} = e^{-\sum_i^k \tau_i} = e^{-\sum_i^k \mu_{t,i}\,\Delta_i}$$

Regular tracking (piecewise constant)



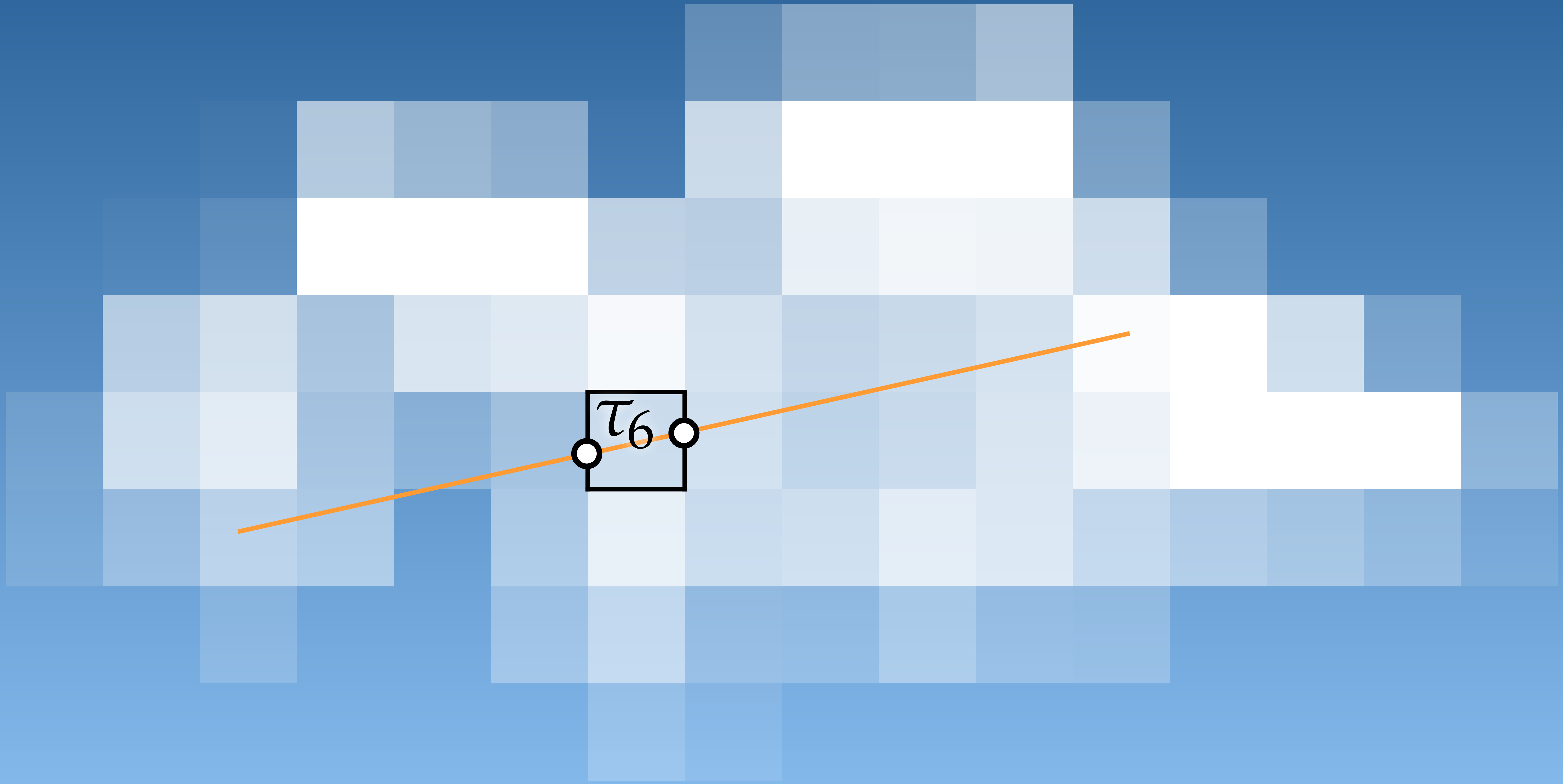
$$T(t) = \mathrm{e}^{-\tau(t)} = \mathrm{e}^{-\sum_i^k \tau_i} = \mathrm{e}^{-\sum_i^k \mu_{\mathrm{t},i}\,\Delta_i}$$

Regular tracking (piecewise constant)



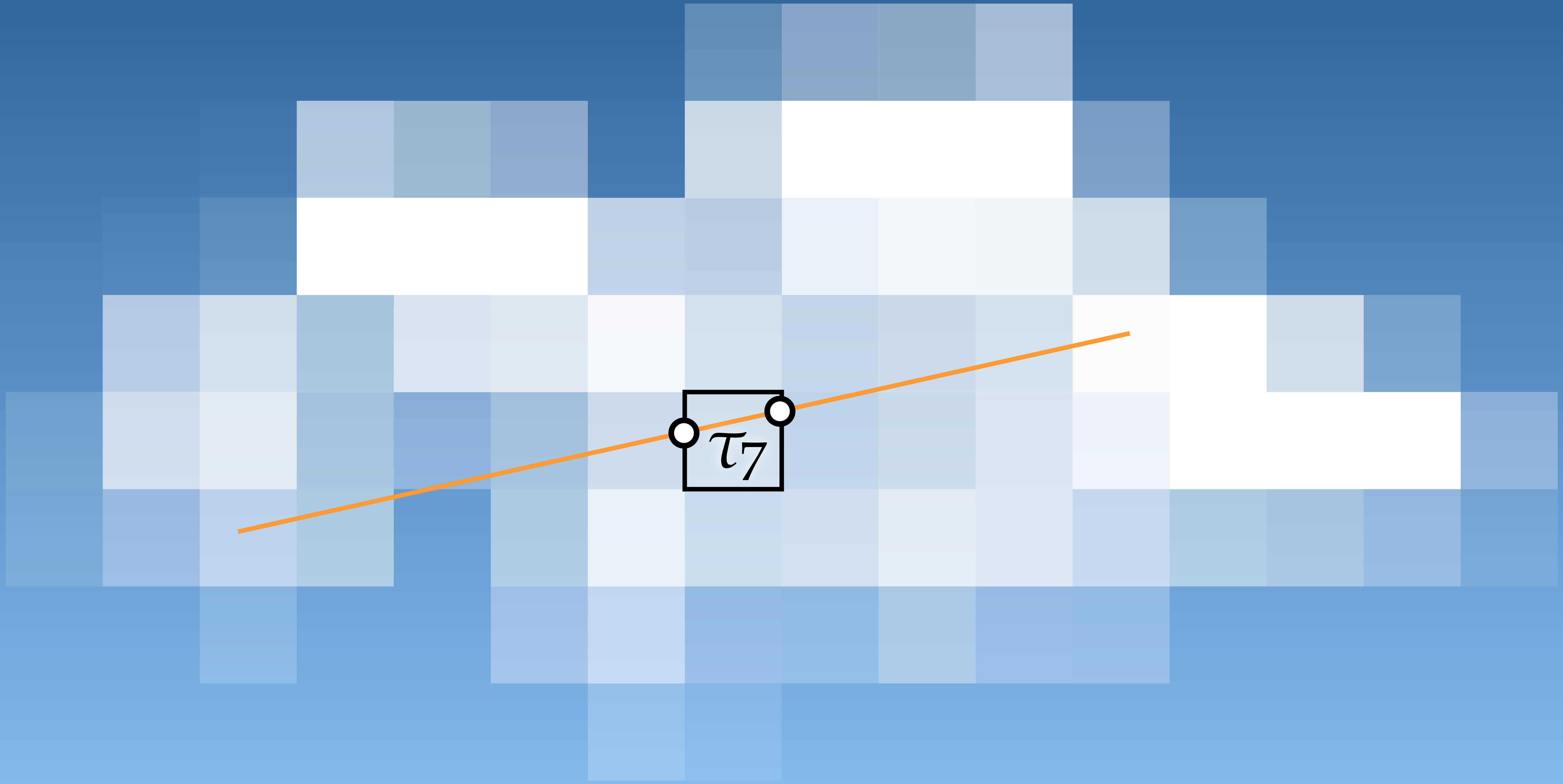
$$T(t) = \mathrm{e}^{-\tau(t)} = \mathrm{e}^{-\sum_i^k \tau_i} = \mathrm{e}^{-\sum_i^k \mu_{\mathrm{t},i}\,\Delta_i}$$

Regular tracking (piecewise constant)



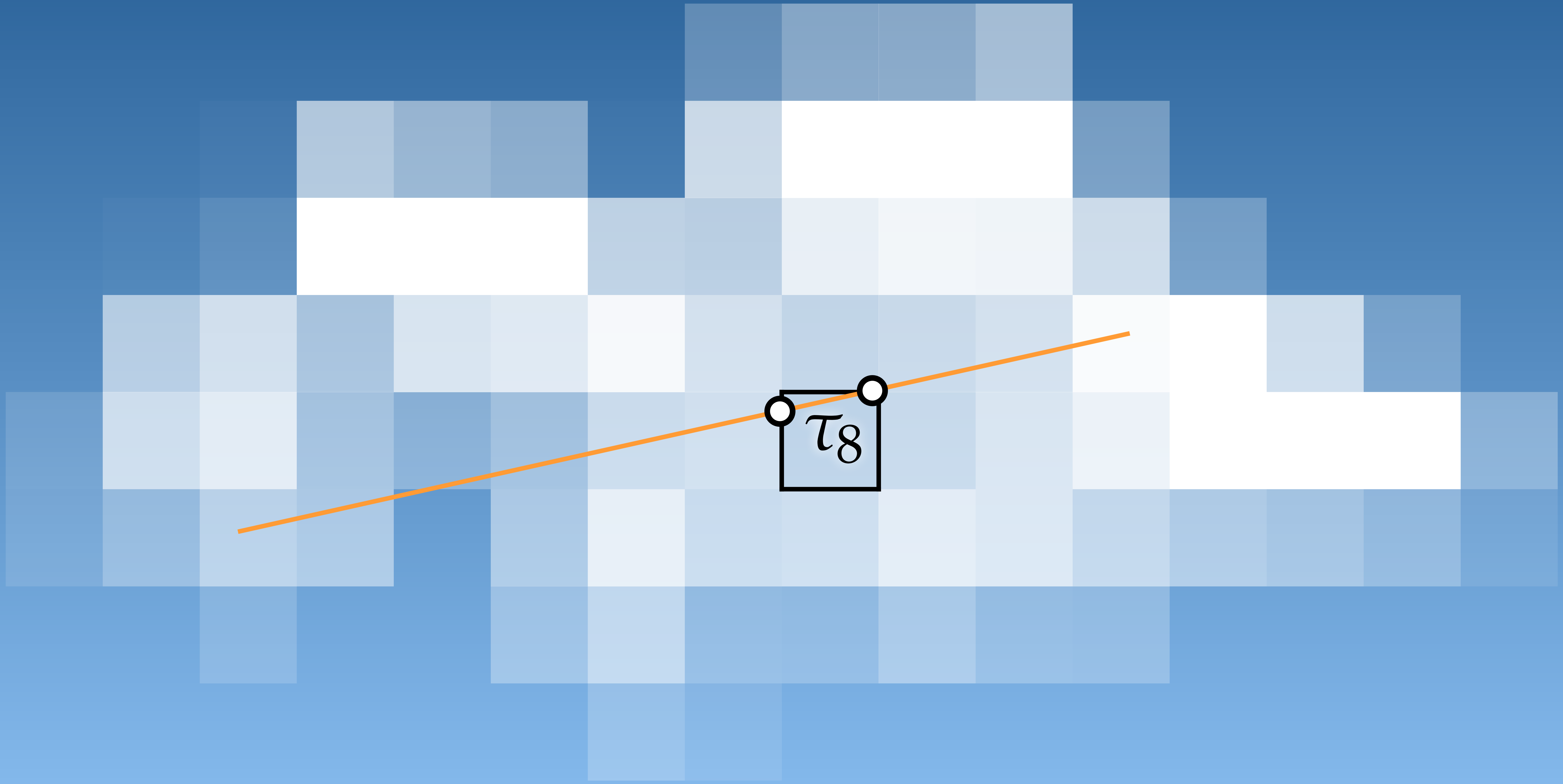
$$T(t) = \mathrm{e}^{-\tau(t)} = \mathrm{e}^{-\sum_i^k \tau_i} = \mathrm{e}^{-\sum_i^k \mu_{\mathrm{t},i}\,\Delta_i}$$

Regular tracking (piecewise constant)



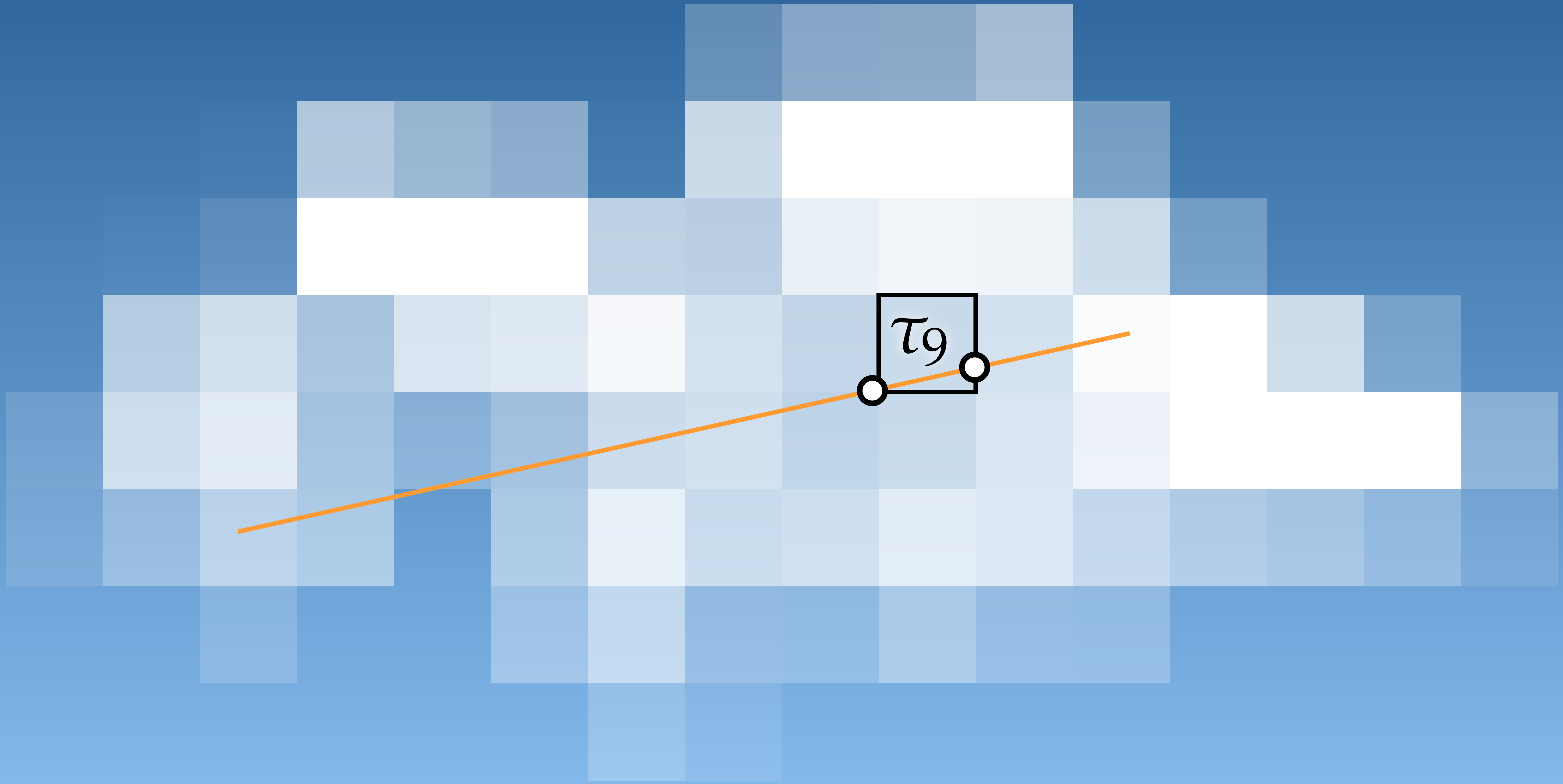
$$T(t) = e^{-\tau(t)} = e^{-\sum_i^k \tau_i} = e^{-\sum_i^k \mu_{t,i}\,\Delta_i}$$

Regular tracking (piecewise constant)



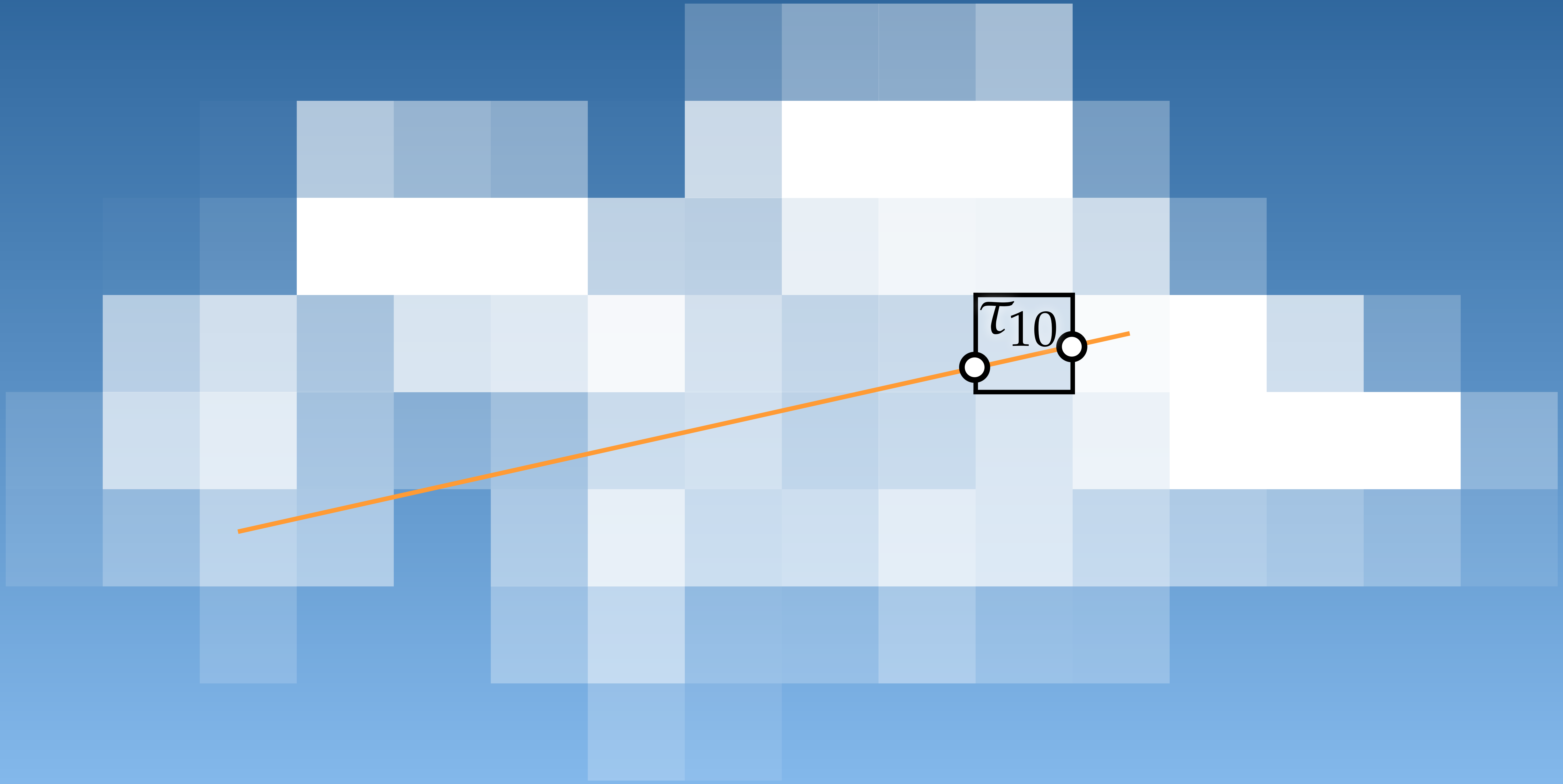
$$T(t) = \mathrm{e}^{-\tau(t)} = \mathrm{e}^{-\sum_i^k \tau_i} = \mathrm{e}^{-\sum_i^k \mu_{\mathrm{t},i}\,\Delta_i}$$

Regular tracking (piecewise constant)



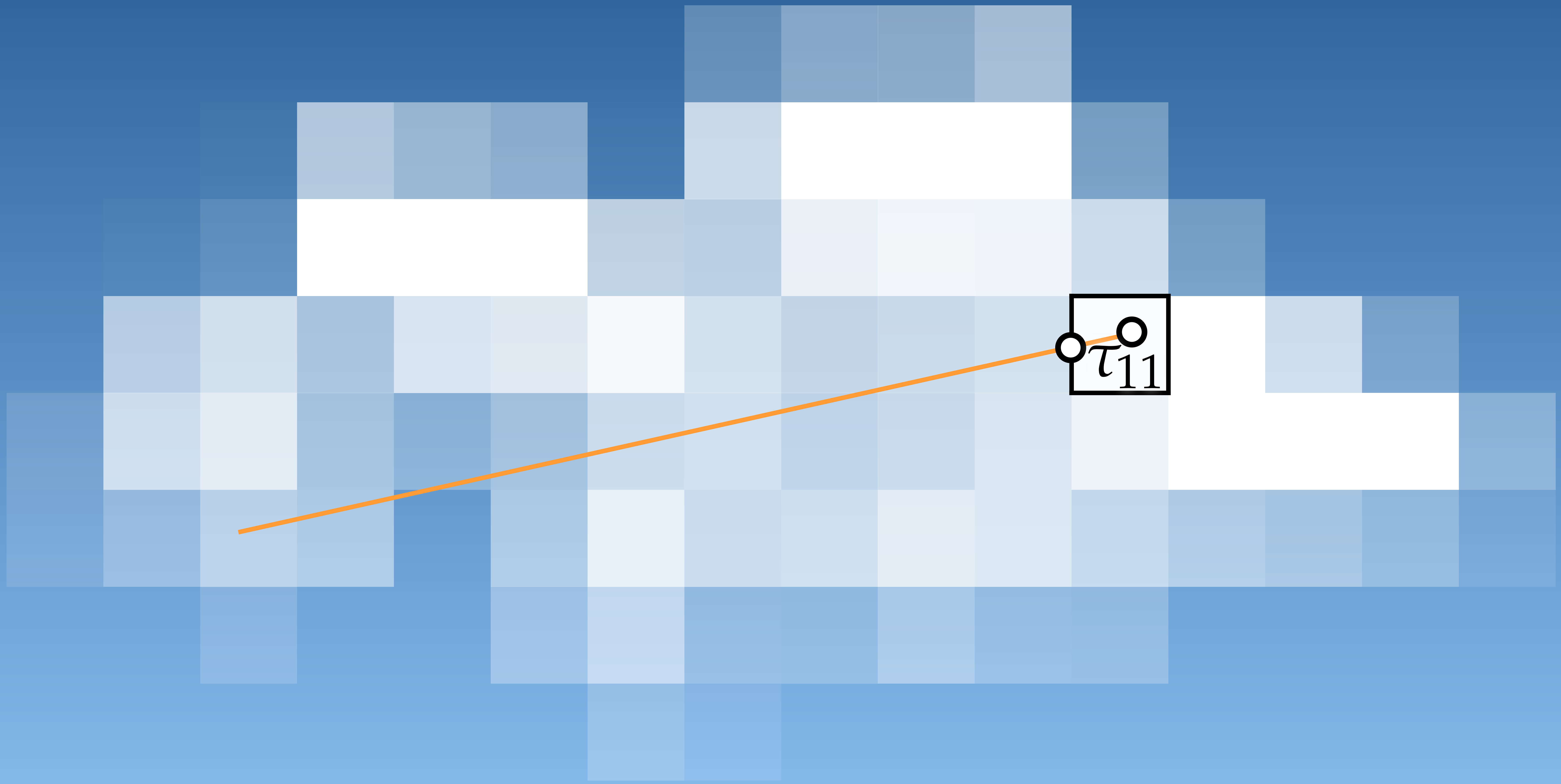
$$T(t) = \mathrm{e}^{-\tau(t)} = \mathrm{e}^{-\sum_i^k \tau_i} = \mathrm{e}^{-\sum_i^k \mu_{\mathrm{t},i}\, \Delta_i}$$

Regular tracking (piecewise constant)



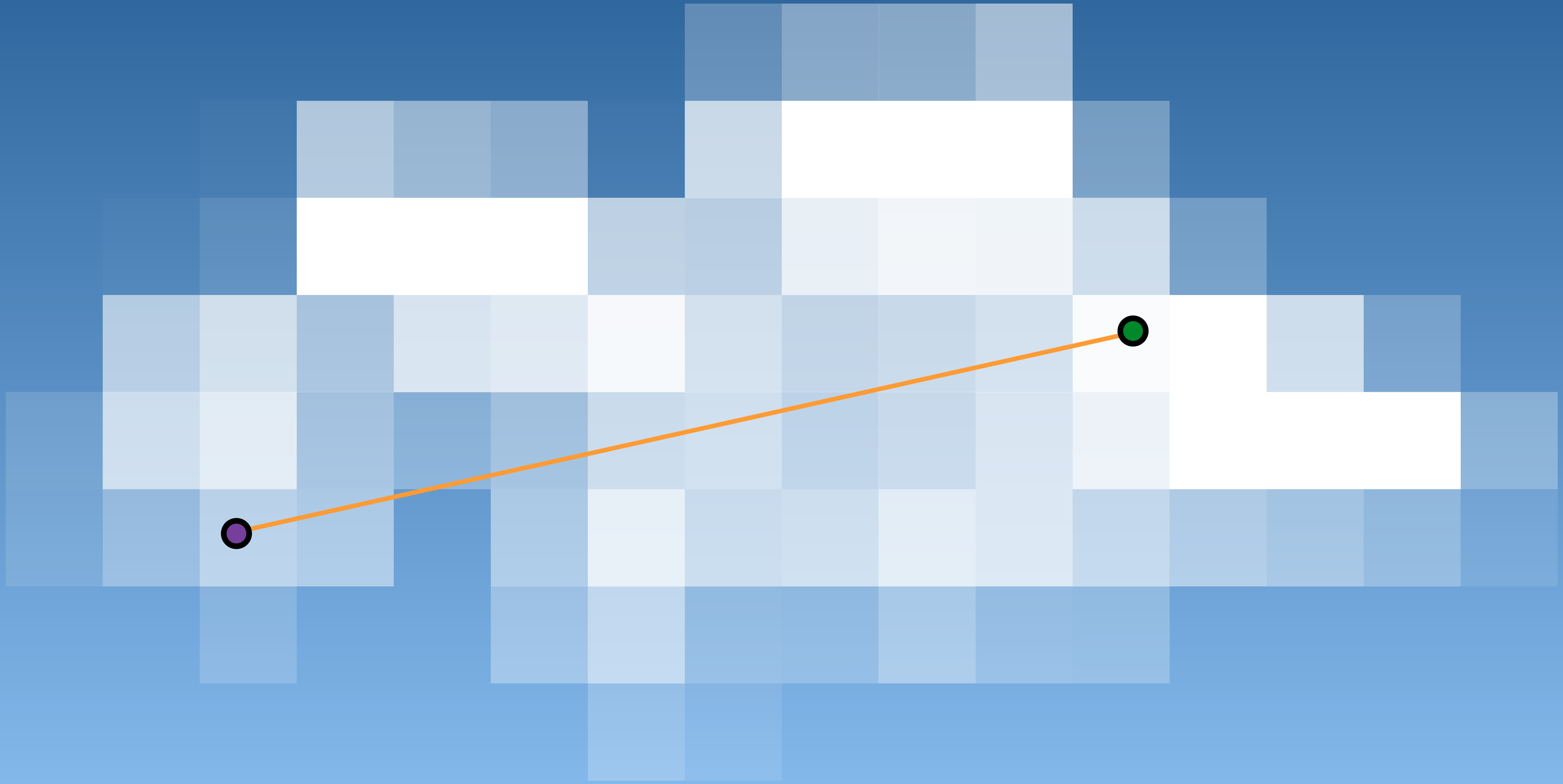
$$T(t) = \mathrm{e}^{-\tau(t)} = \mathrm{e}^{-\sum_i^k \tau_i} = \mathrm{e}^{-\sum_i^k \mu_{\mathrm{t},i}\,\Delta_i}$$

Regular tracking (piecewise constant)



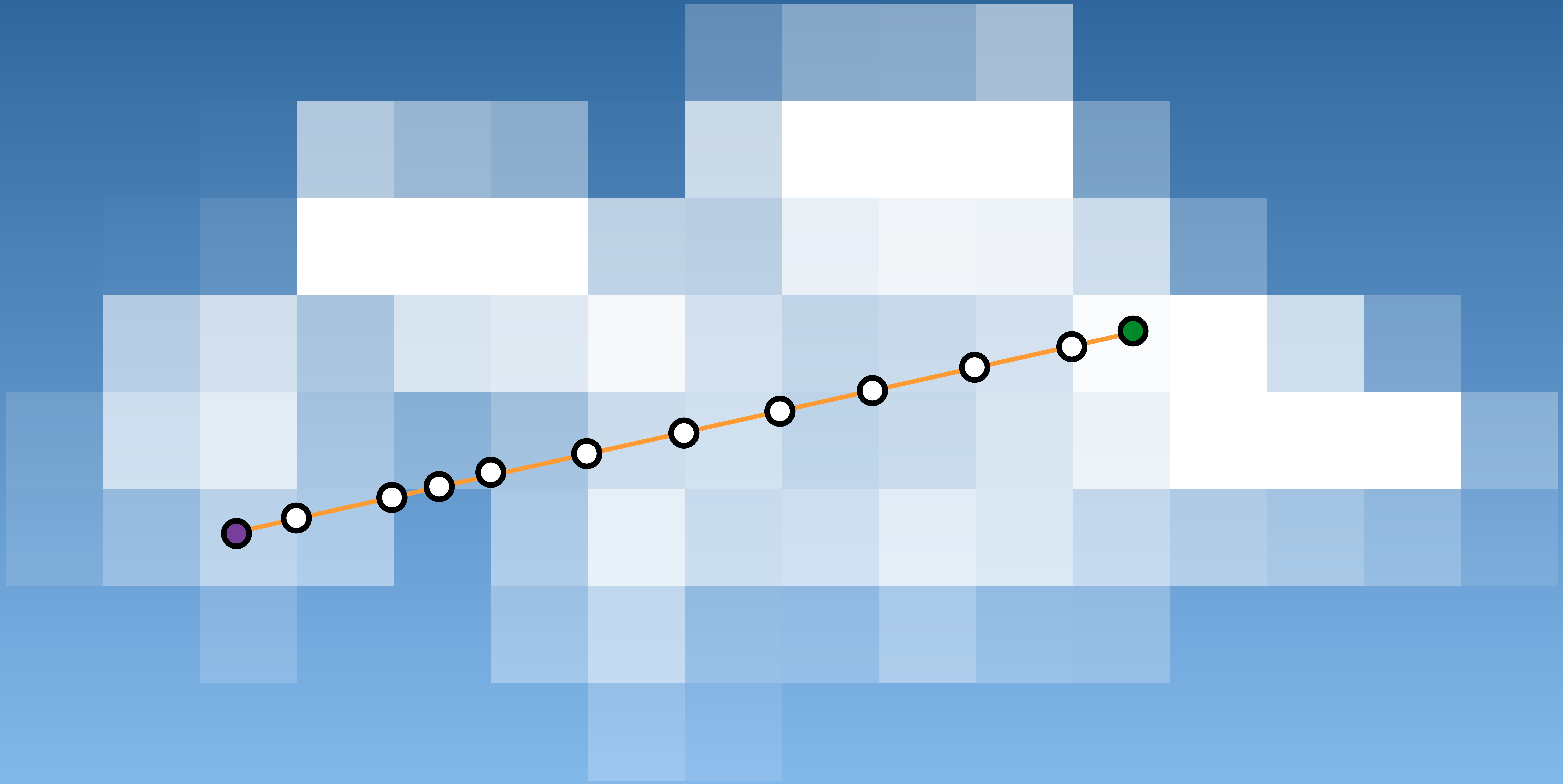
$$T(t) = \mathrm{e}^{-\tau(t)} = \mathrm{e}^{-\sum_i^k \tau_i} = \mathrm{e}^{-\sum_i^k \mu_{\mathrm{t},i}\,\Delta_i}$$

Regular tracking (piecewise constant)



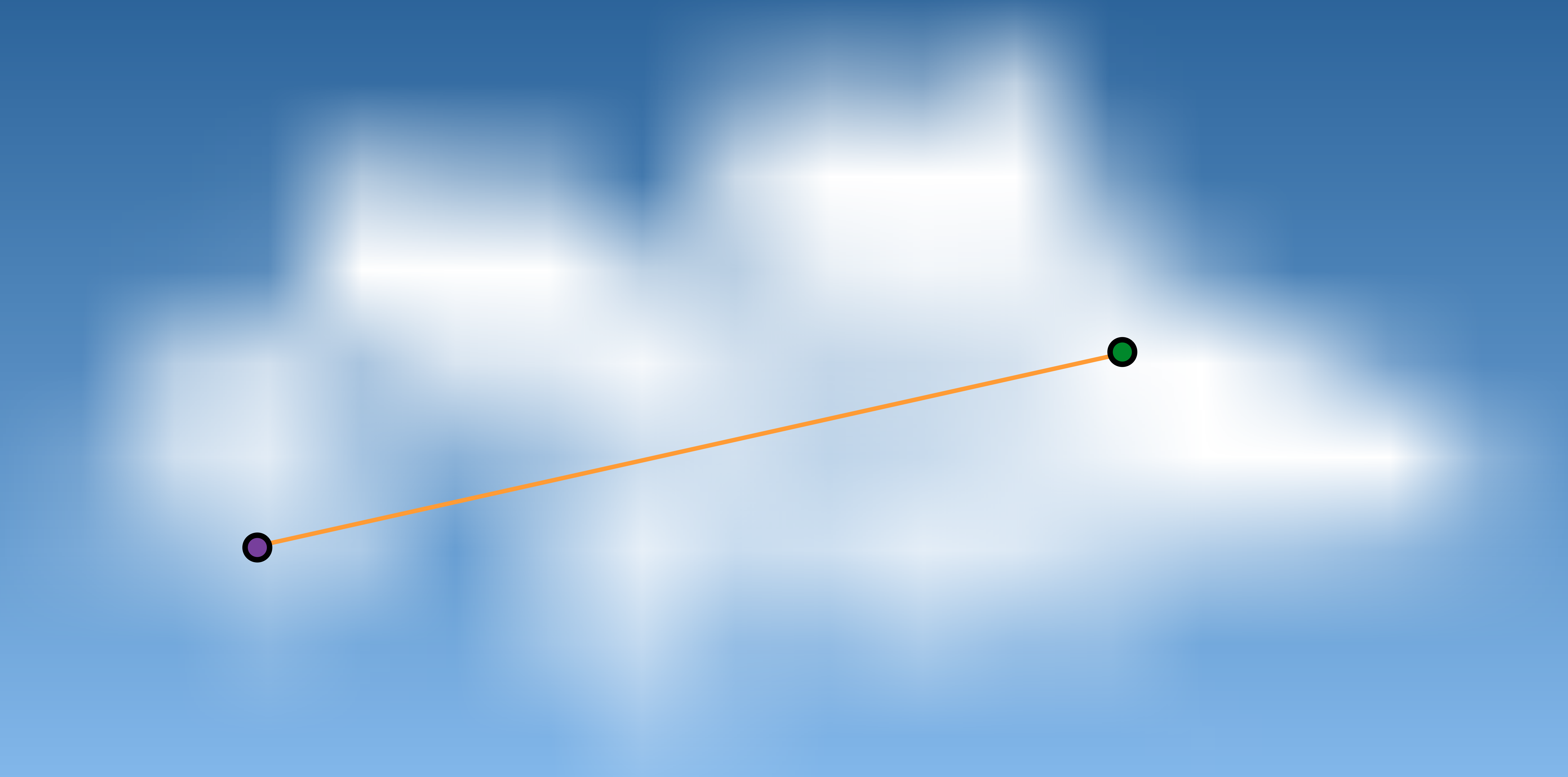
$$T(t) = \mathrm{e}^{-\tau(t)} = \mathrm{e}^{-\sum_i^k \tau_i} = \mathrm{e}^{-\sum_i^k \mu_{\mathrm{t},i}\,\Delta_i}$$

Regular tracking (piecewise constant)



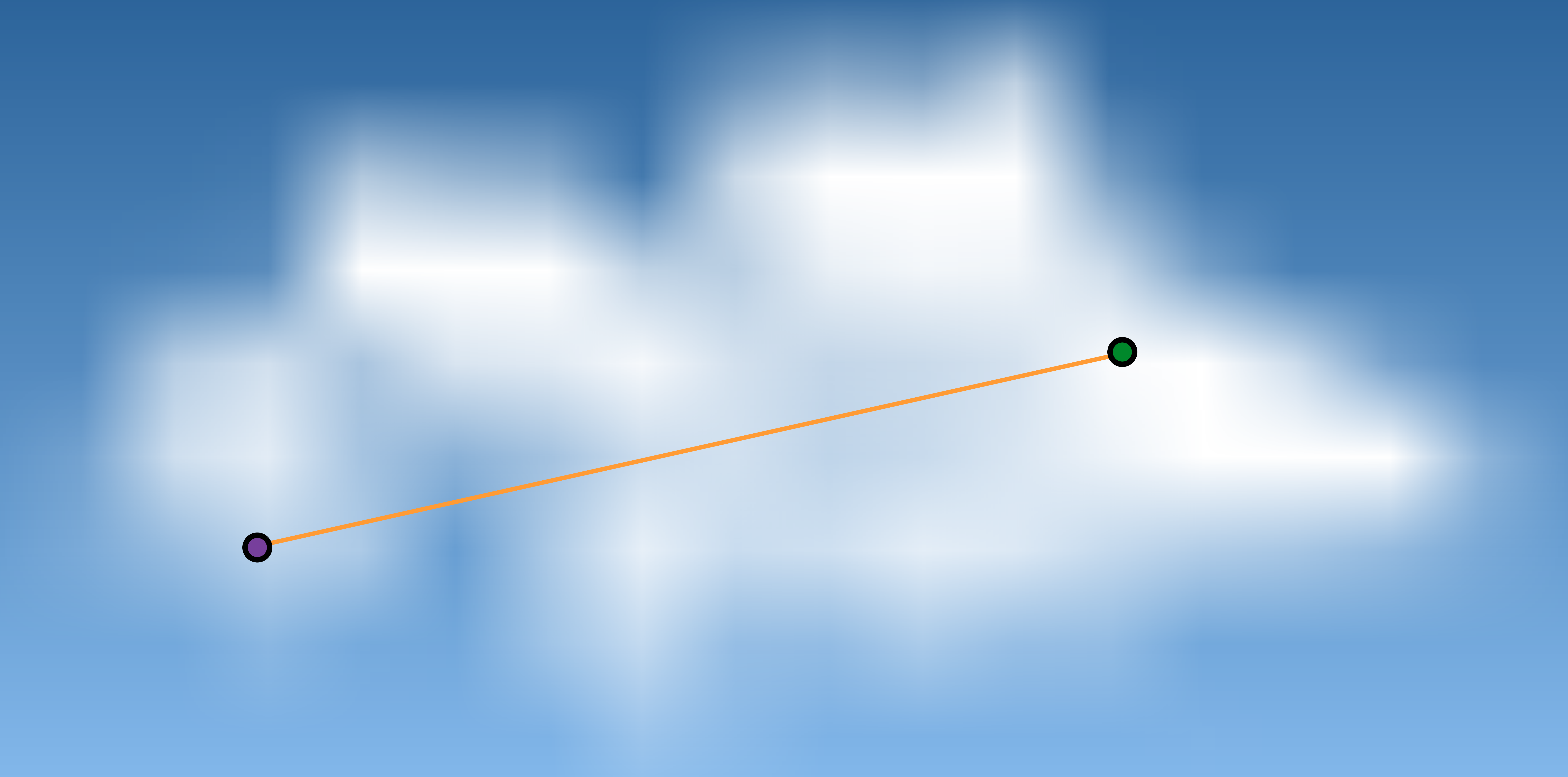
$$T(t) = e^{-\tau(t)} = e^{-\sum_i^k \tau_i} = e^{-\sum_i^k \mu_{t,i}\,\Delta_i}$$

Regular tracking (bi-linear/tri-linear)



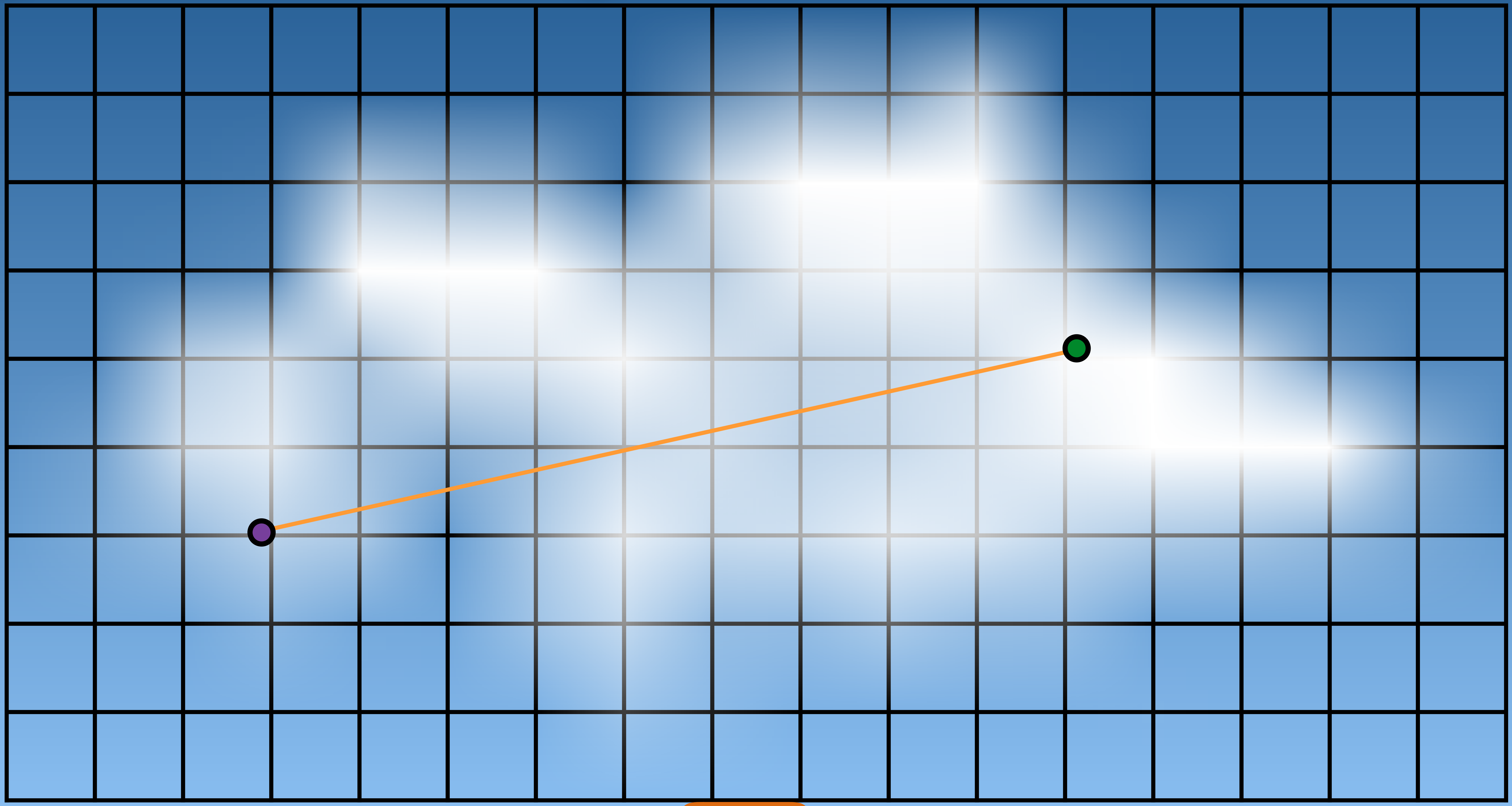
$$T(t) = e^{-\tau(t)} = e^{-\sum_i^k \tau_i} =$$

Regular tracking (bi-linear/tri-linear)



$$T(t) = \mathrm{e}^{-\tau(t)} = \mathrm{e}^{-\boxed{\sum_i^k \tau_i}} = \text{analytic, but more complicated}$$

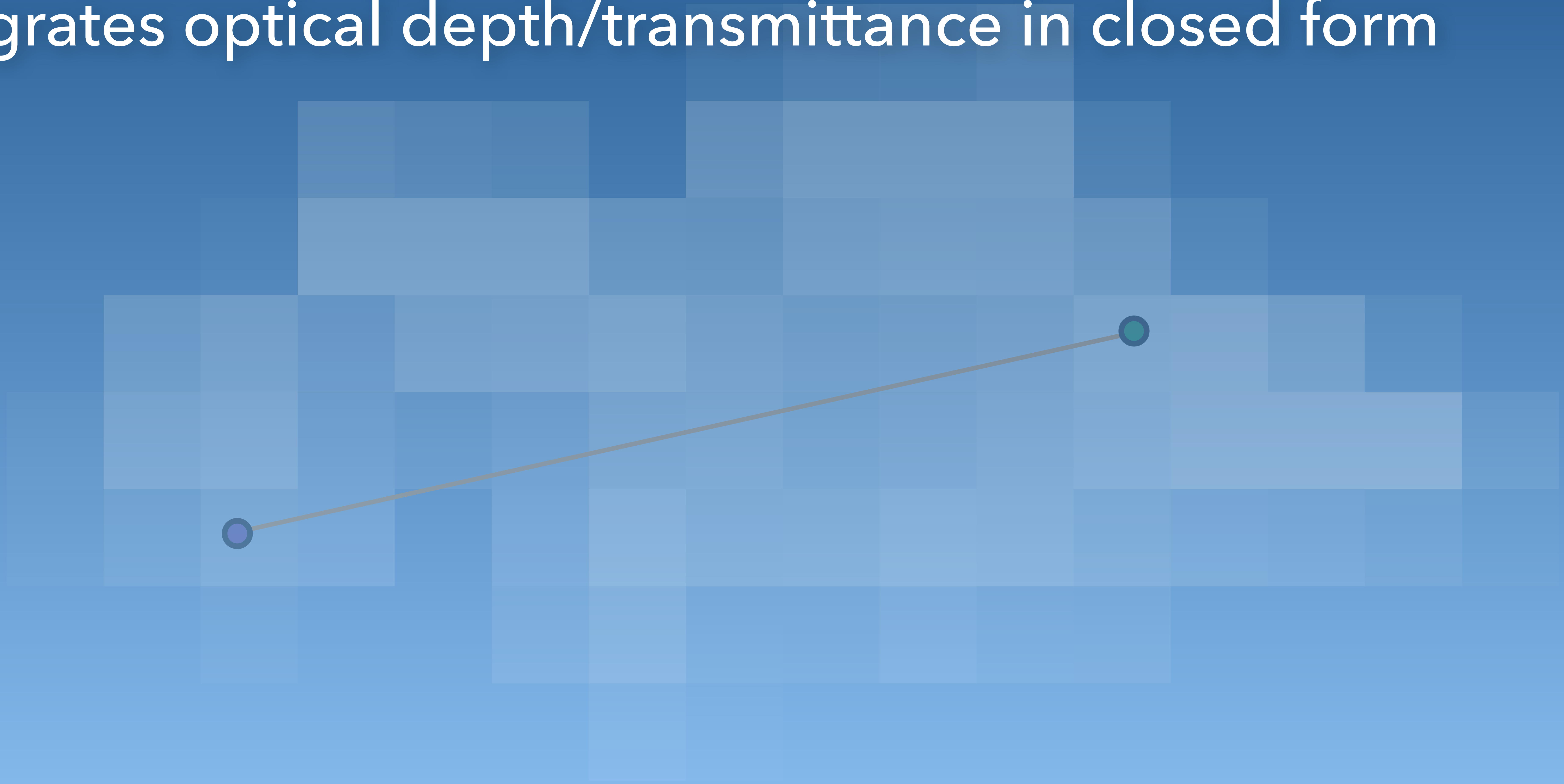
Regular tracking (bi-linear/tri-linear)



$$T(t) = \mathrm{e}^{-\tau(t)} = \mathrm{e}^{-\sum_i^k \tau_i} = \text{analytic, but more complicated}$$

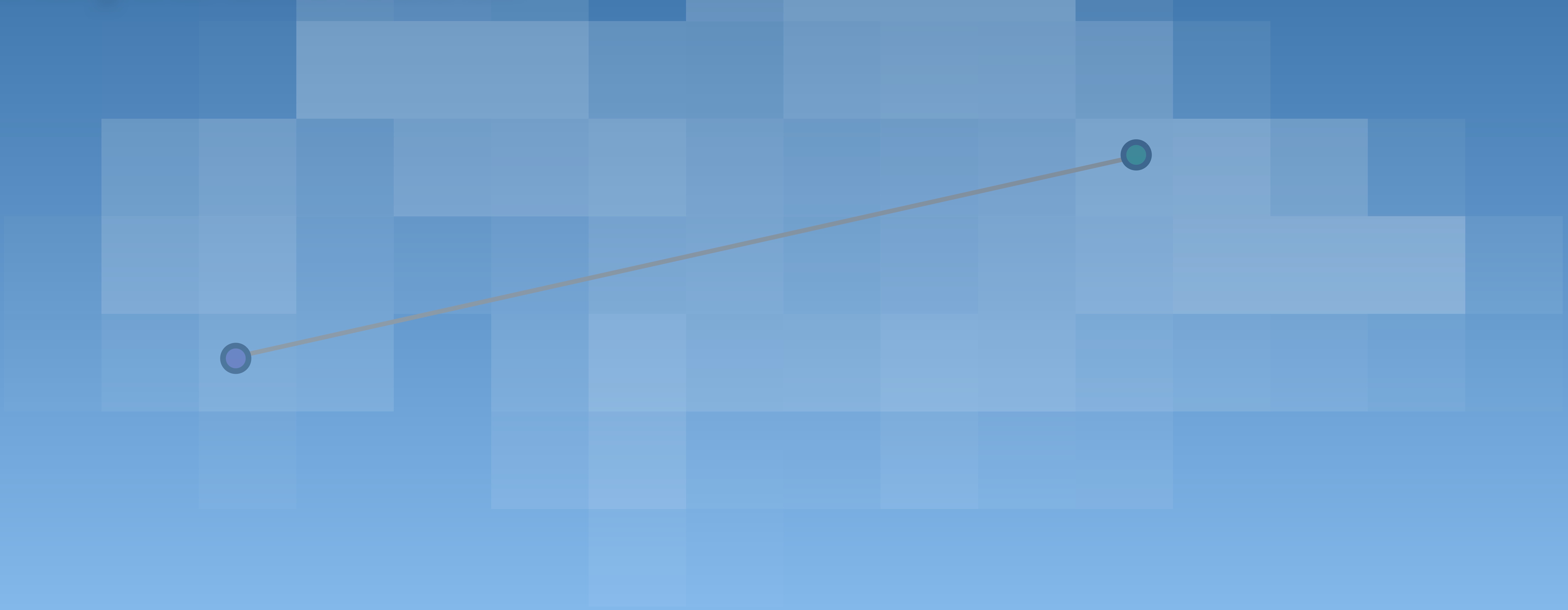
Regular tracking

✔ Integrates optical depth/transmittance in closed form



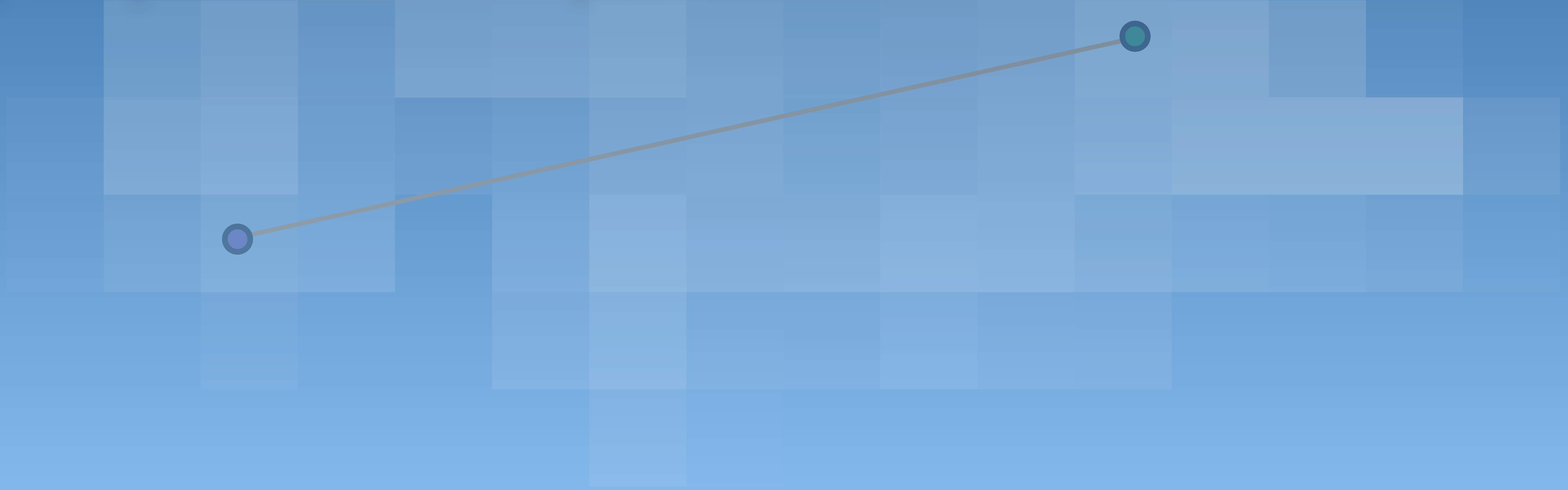
Regular tracking

- ✔ Integrates optical depth/transmittance in closed form
  - ✔ Analytic/unbiased!



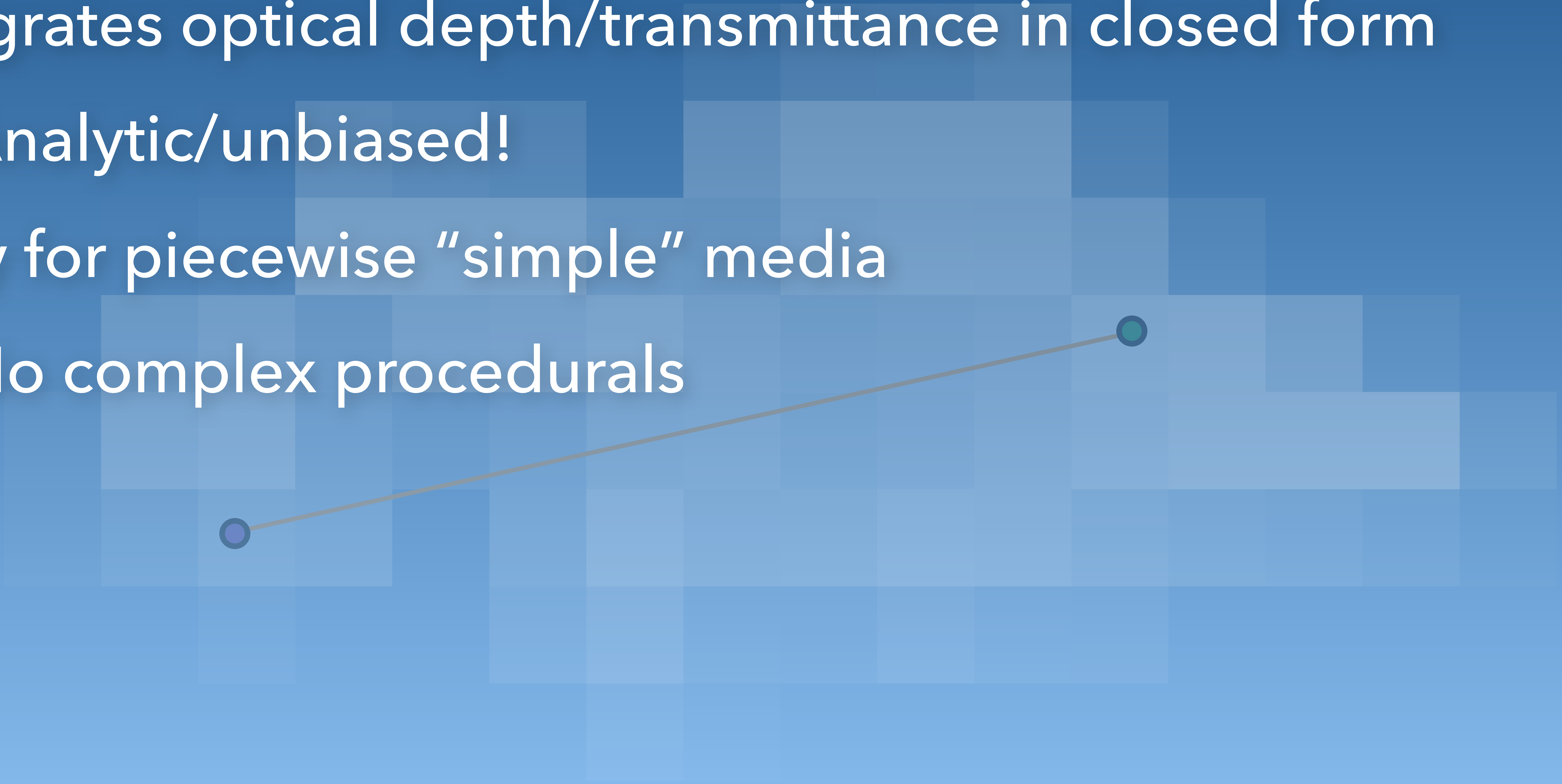
Regular tracking

- ✔ Integrates optical depth/transmittance in closed form
    - ✔ Analytic/unbiased!
- ✘ Only for piecewise “simple” media



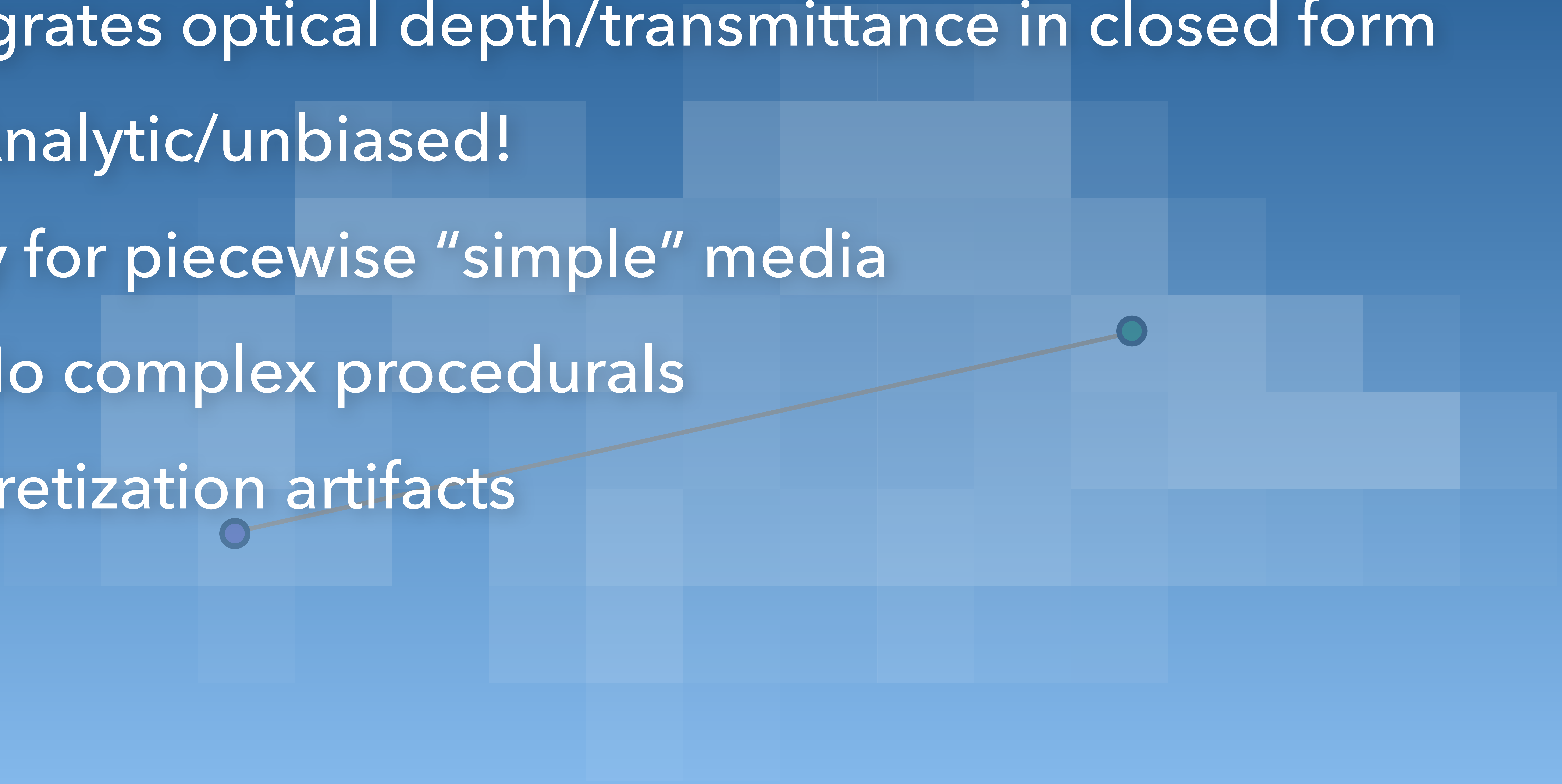
Regular tracking

- ✔ Integrates optical depth/transmittance in closed form
  - ✔ Analytic/unbiased!
- ✘ Only for piecewise “simple” media
  - ✘ No complex procedurals



Regular tracking

- ✔ Integrates optical depth/transmittance in closed form
  - ✔ Analytic/unbiased!
- ✘ Only for piecewise “simple” media
  - ✘ No complex procedurals
- ✘ Discretization artifacts



Regular tracking

- ✔ Integrates optical depth/transmittance in closed form
  - ✔ Analytic/unbiased!
- ✘ Only for piecewise “simple” media
  - ✘ No complex procedurals
- ✘ Discretization artifacts
- ✘ Expensive

Regular tracking

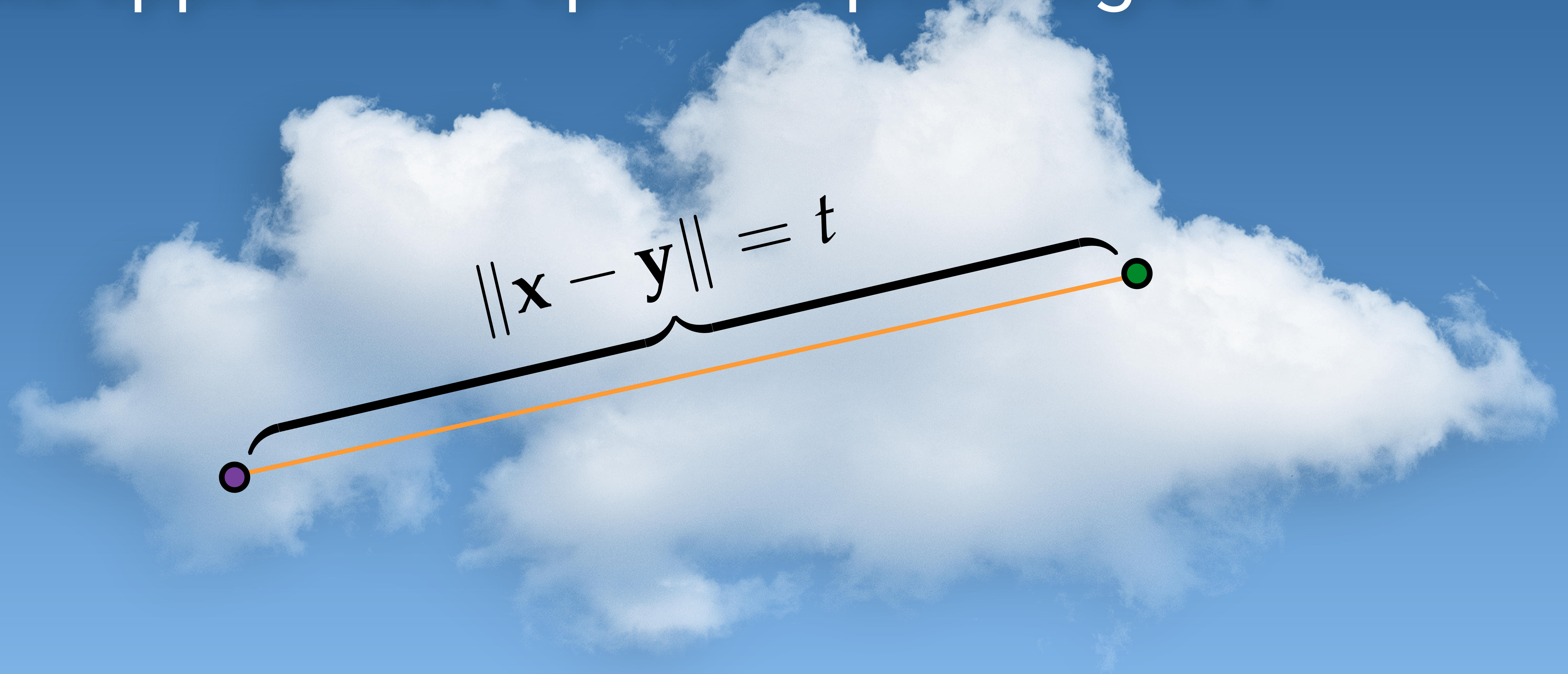
- ✔ Integrates optical depth/transmittance in closed form
  - ✔ Analytic/unbiased!
- ✘ Only for piecewise “simple” media
  - ✘ No complex procedurals
- ✘ Discretization artifacts
- ✘ Expensive
- ✘ No speed/quality knob

Regular tracking

- ✔ Integrates optical depth/transmittance in closed form
  - ✔ Analytic/unbiased!
- ✘ Only for piecewise “simple” media
  - ✘ No complex procedurals
- ✘ Discretization artifacts
- ✘ Expensive
- ✘ No speed/quality knob
- ✘ Not progressive

Integrating τ

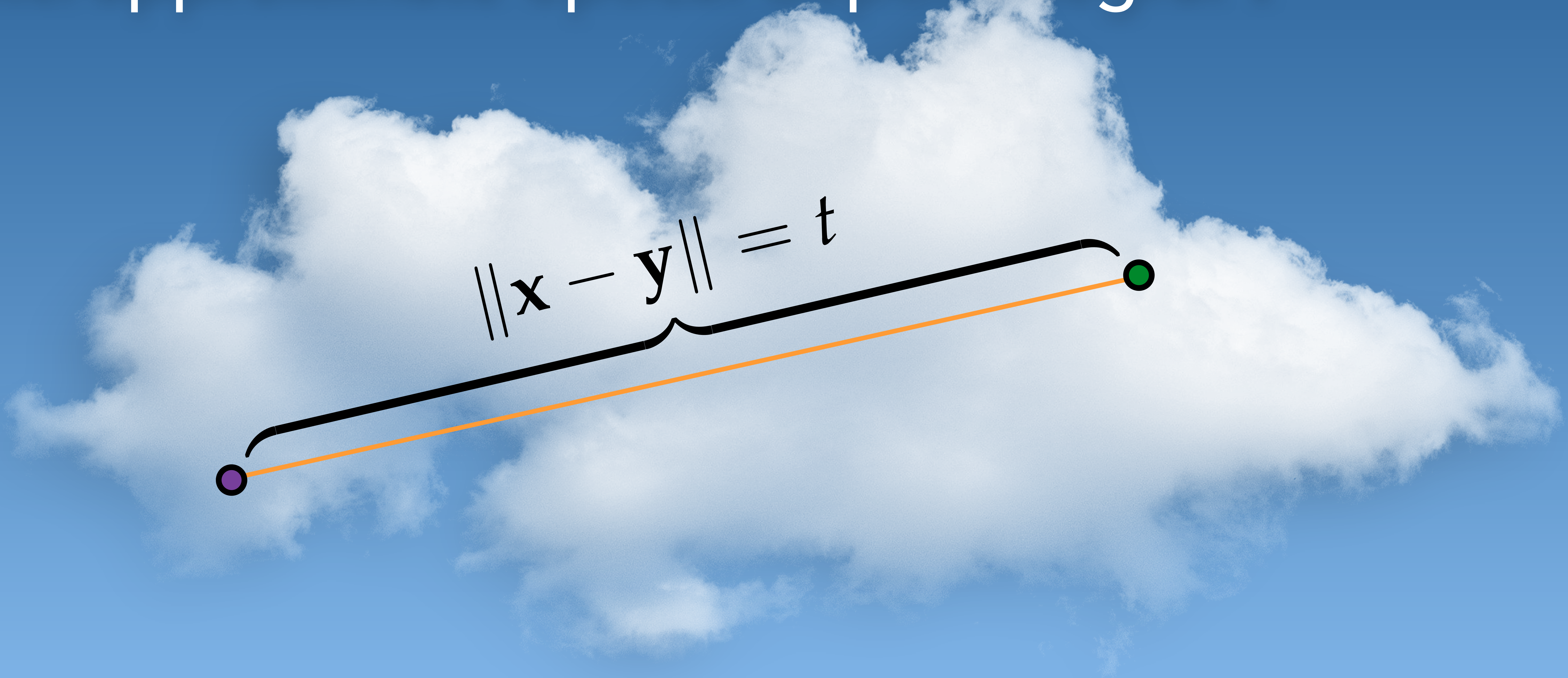
Estimate/approximate optical depth integral τ



$$\tau(t) = \int_0^t \mu_t(t')\,\mathrm{d}t'$$

Integrating τ

Estimate/approximate optical depth integral τ



$$\langle T(t) \rangle_{\mathrm{RM}} = \mathrm{e}^{-\langle \tau(t) \rangle} \qquad \tau(t) = \int_0^t \mu_{\mathrm{t}}(t') \, \mathrm{d}t'$$

Ray marching (Quadrature)

$$\langle T(t) \rangle_{\mathrm{RM}} = \mathrm{e}^{-\langle \tau(t) \rangle}$$

$$\langle \tau(t) \rangle_{\mathrm{RS}} = \sum_{i=1}^{k} \mu_{\mathrm{t}}(t_i)\, \Delta t_i \approx \int_0^t \mu_{\mathrm{t}}(t')\, \mathrm{d}t'$$

Ray marching (Quadrature)

Riemann sum

$$\langle T(t)\rangle_{\mathrm{RM}} = \mathrm{e}^{-\langle \tau(t)\rangle}$$

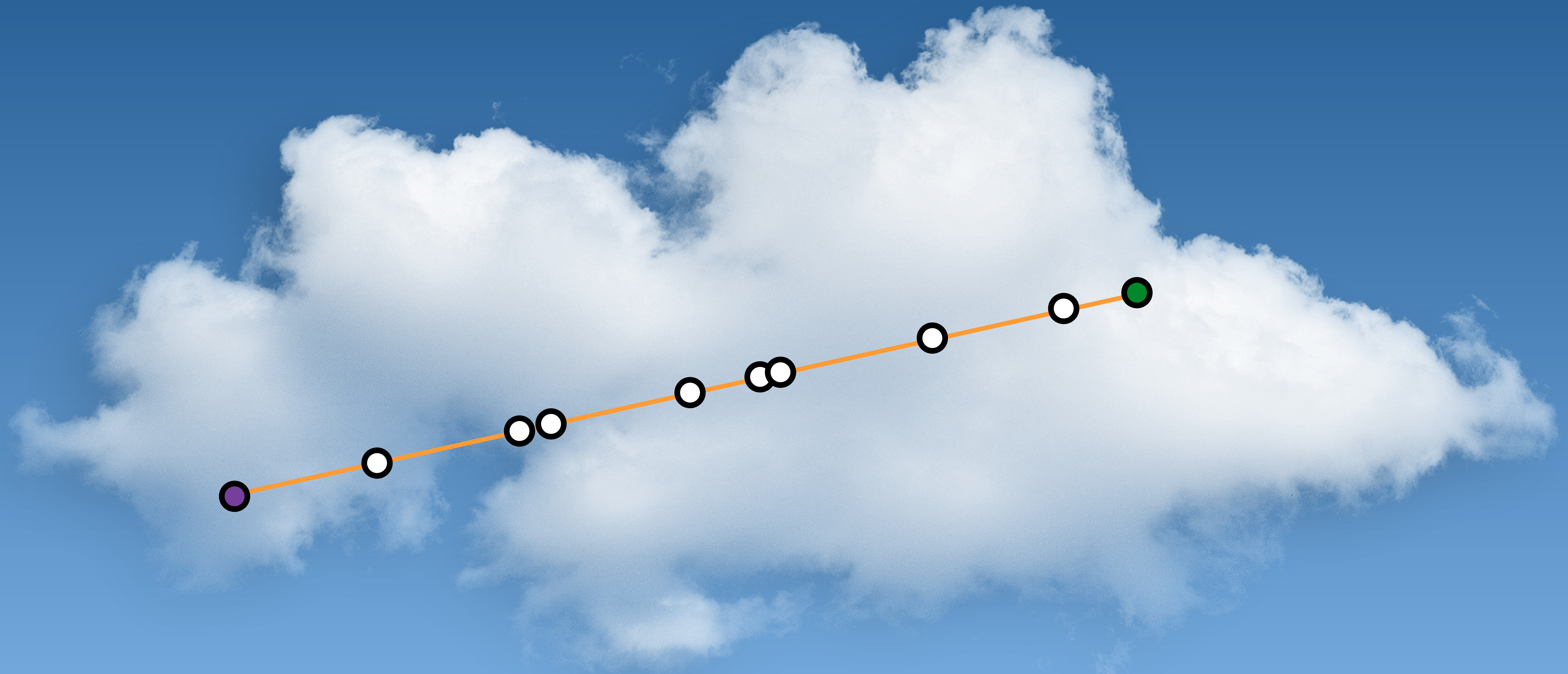
$$\langle \tau(t)\rangle_{\mathrm{RS}} = \sum_{i=1}^{k} \mu_{\mathrm{t}}(t_i)\,\Delta t_i \approx \int_0^t \mu_{\mathrm{t}}(t')\,\mathrm{d}t'$$

Ray marching (Monte Carlo)

$$\langle T(t) \rangle_{\mathrm{RM}} = \mathrm{e}^{-\langle \tau(t) \rangle}$$

$$\langle \tau(t) \rangle_{\mathrm{MC}} = \sum_{i=1}^{k} \frac{\mu_{\mathrm{t}}(t_i)}{p(t_i)k} \quad \text{with } t_i \propto p(t_i)$$

Ray marching (Monte Carlo)



$$\langle T(t)\rangle_{\mathrm{RM}} = \mathrm{e}^{-\langle \tau(t)\rangle}$$

$$\langle \tau(t)\rangle_{\mathrm{MC}} = \sum_{i=1}^{k} \frac{\mu_{\mathrm{t}}(t_i)}{p(t_i)k} \quad \text{with } t_i \propto p(t_i)$$

Ray marching (Stratified Monte Carlo)

$$\langle T(t)\rangle_{\mathrm{RM}} = \mathrm{e}^{-\langle \tau(t)\rangle}$$

$$\langle \tau(t)\rangle_{\mathrm{SMC}} = \sum_{i=1}^{k} \mu_{\mathrm{t}}(t_i)\Delta_t, \quad \text{with} \quad t_i = (i-\xi)\Delta_t$$

Ray marching (Stratified Monte Carlo)

$$\langle T(t)\rangle_{\mathrm{RM}} = \mathrm{e}^{-\langle \tau(t)\rangle}$$

$$\langle \tau(t)\rangle_{\mathrm{SMC}} = \sum_{i=1}^{k} \mu_{\mathrm{t}}(t_i)\Delta_t, \quad \text{with} \quad t_i = (i-\xi)\Delta_t$$

Ray marching

✔ Medium not discretized, supports procedurals

Ray marching

- ✔ Medium not discretized, supports procedurals
- ✔ Useful speed/quality knob, and progressive, but…

Ray marching

- ✔ Medium not discretized, supports procedurals
- ✔ Useful speed/quality knob, and progressive, but…
- ✘ Has error: variance/banding/noise, and

Ray marching

- ✔ Medium not discretized, supports procedurals
- ✔ Useful speed/quality knob, and progressive, but…
- ✘ Has error: variance/banding/noise, and
- ✘ **Bias** (not even consistent when used progressively)!

Ray marching

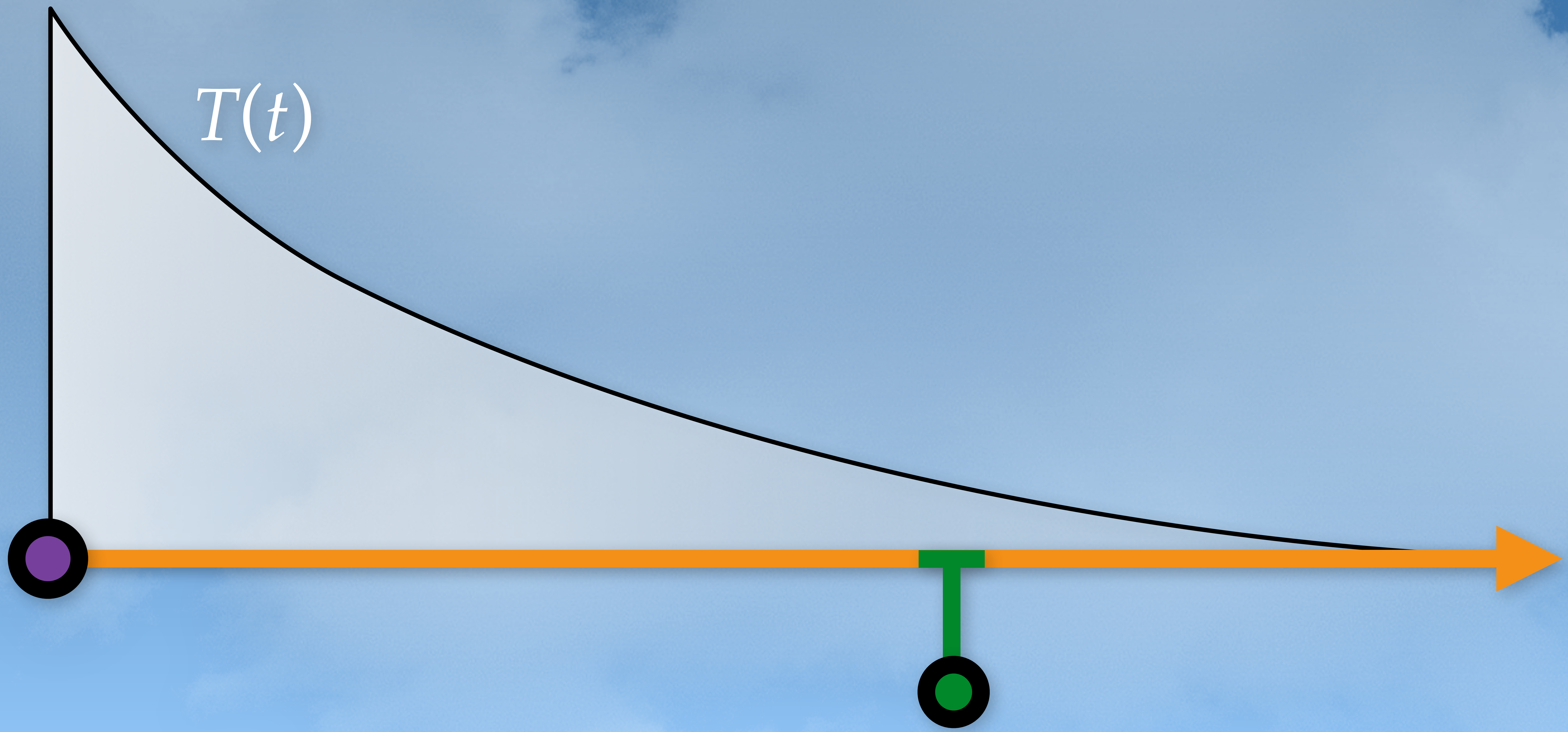
- ✔ Medium not discretized, supports procedurals
- ✔ Useful speed/quality knob, and progressive, but…
- ✘ Has error: variance/banding/noise, and
- ✘ **Bias** (not even consistent when used progressively)!
  - ✘ *unbiased* estimate of optical depth leads to *biased* estimate of transmittance since: $\mathrm{E}\left[\mathrm{e}^{\mathrm{X}}\right] \neq \mathrm{e}^{\mathrm{E}[\mathrm{X}]}$

Ray marching

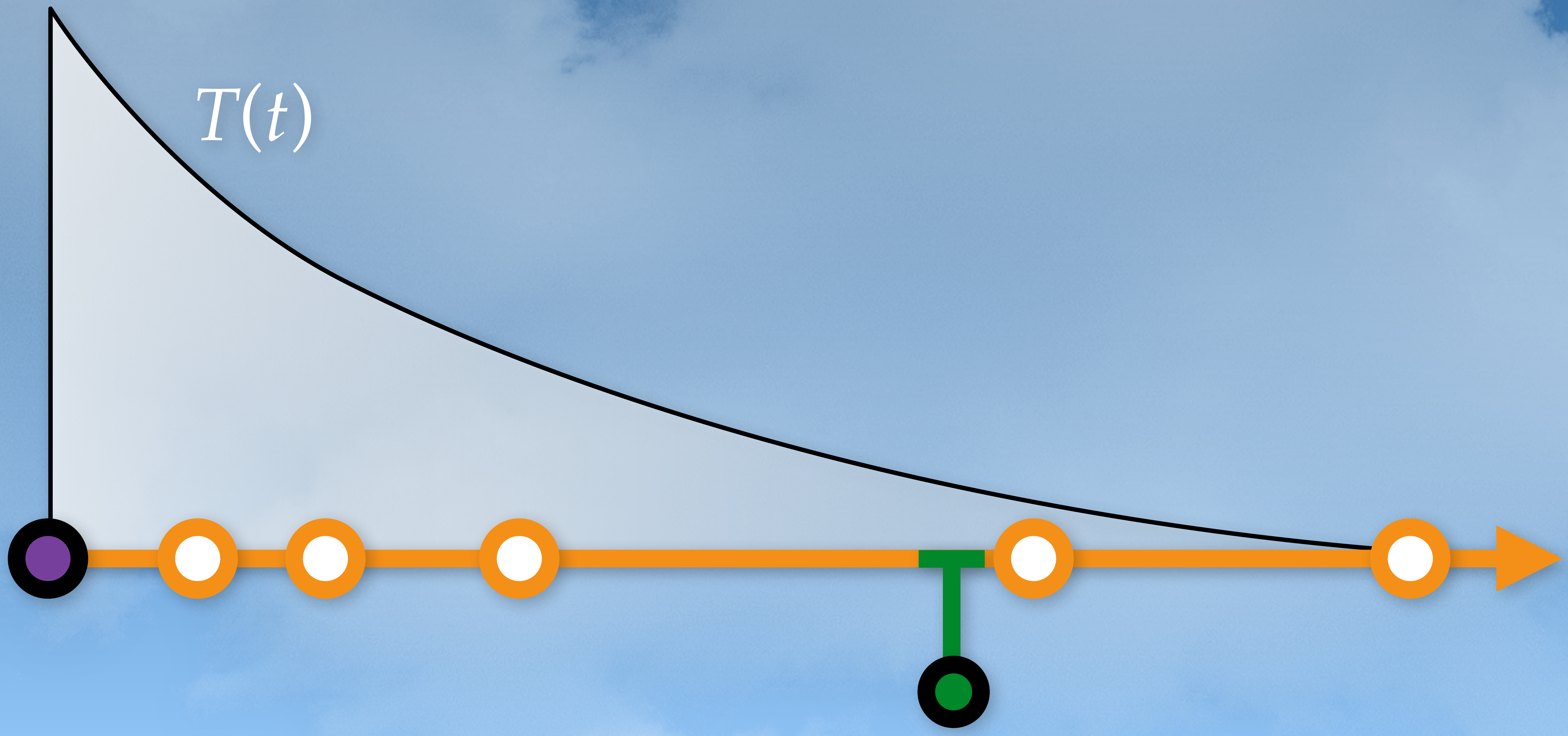
- ✔ Medium not discretized, supports procedurals
- ✔ Useful speed/quality knob, and progressive, but…
- ✘ Has error: variance/banding/noise, and
- ✘ **Bias** (not even consistent when used progressively)!
    - ✘ *unbiased* estimate of optical depth leads to *biased* estimate of transmittance since: $\mathrm{E}\left[\mathrm{e}^{X}\right] \neq \mathrm{e}^{\mathrm{E}[X]}$
    - ✘ Overestimates transmittances (medium looks “thinner” when using large steps)

Transmittance from free-flight sampling

Transmittance from free-flight sampling



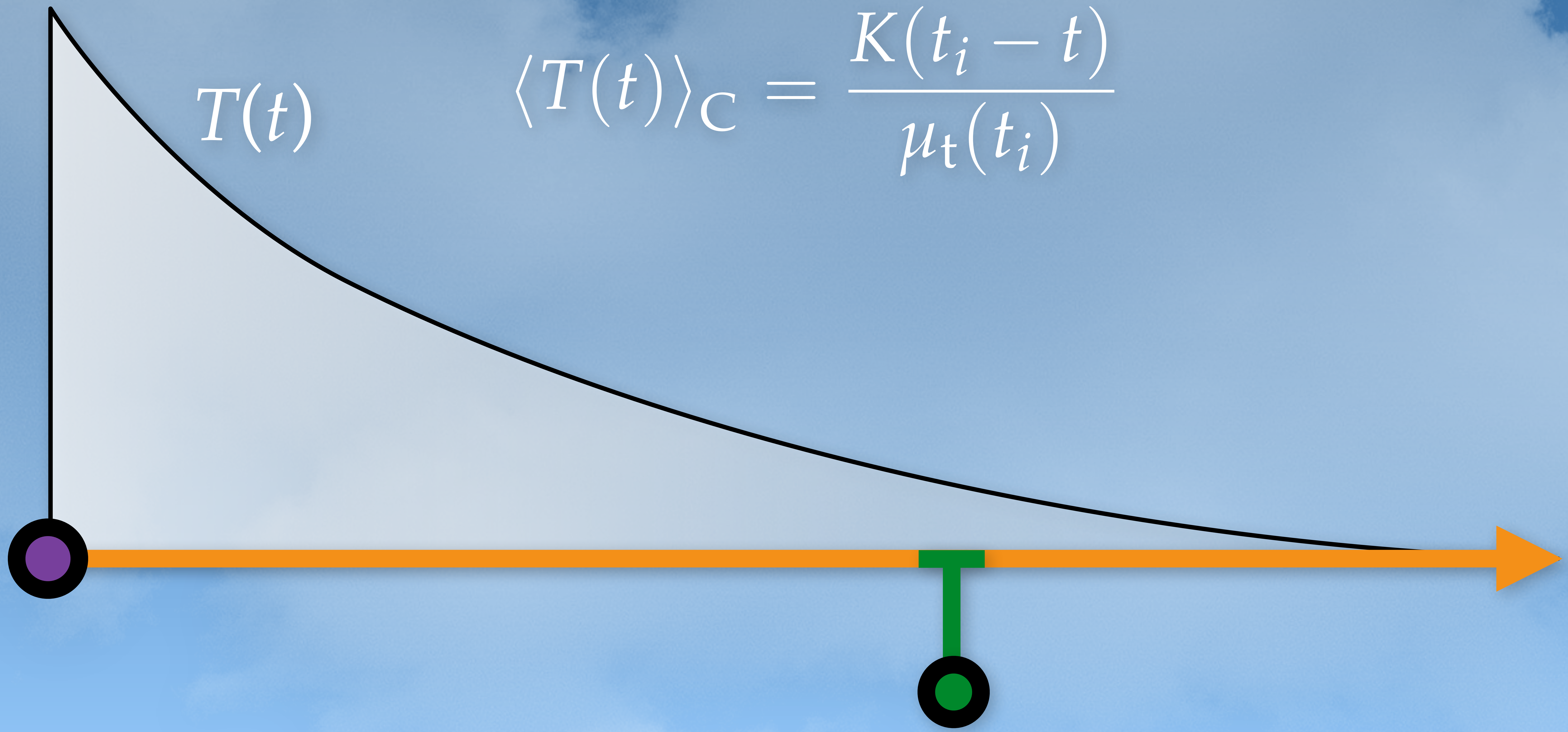
Transmittance from free-flight sampling



Counting free-flights

Collision estimator:

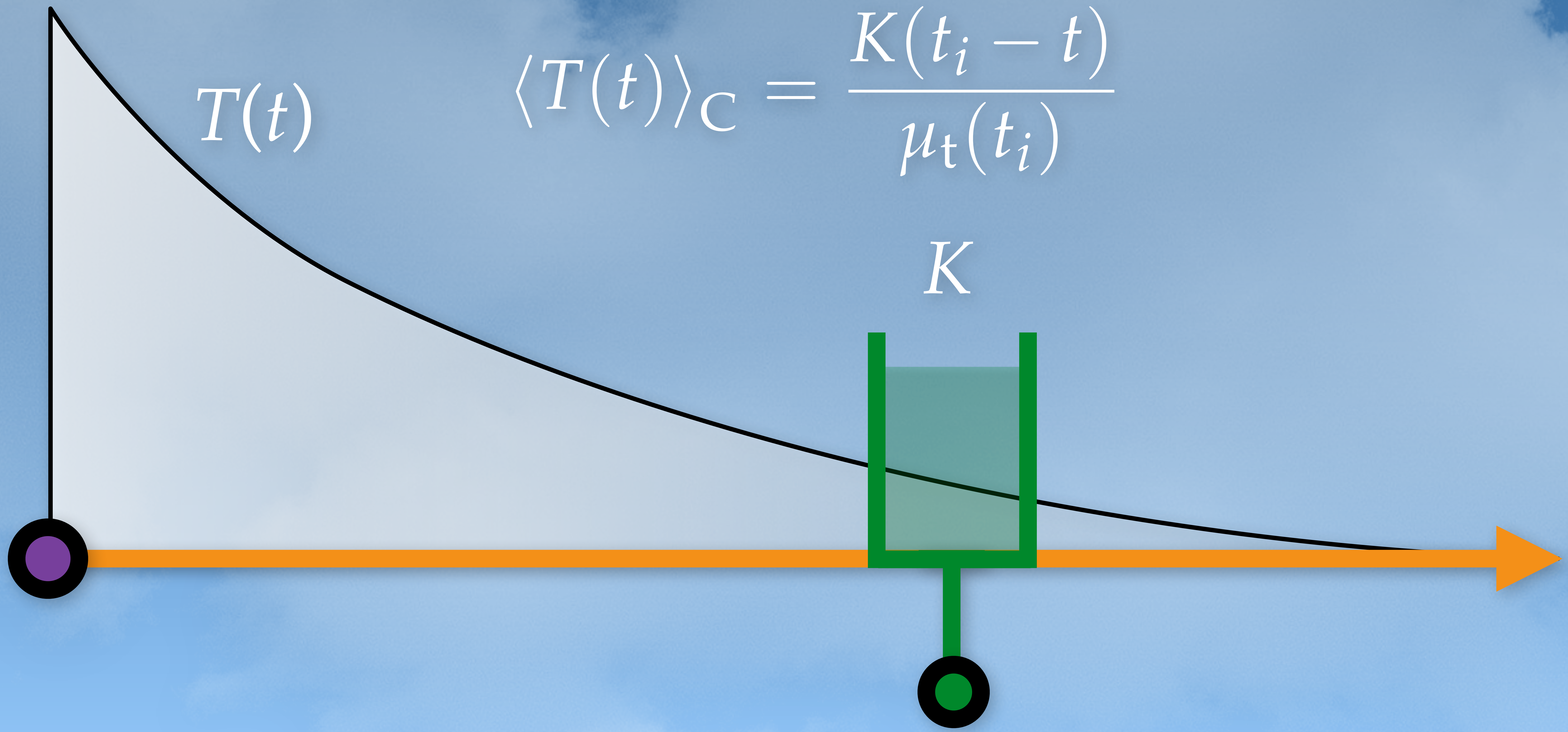
$$\langle T(t) \rangle_C = \frac{K(t_i - t)}{\mu_t(t_i)}$$



Counting free-flights

Collision estimator:

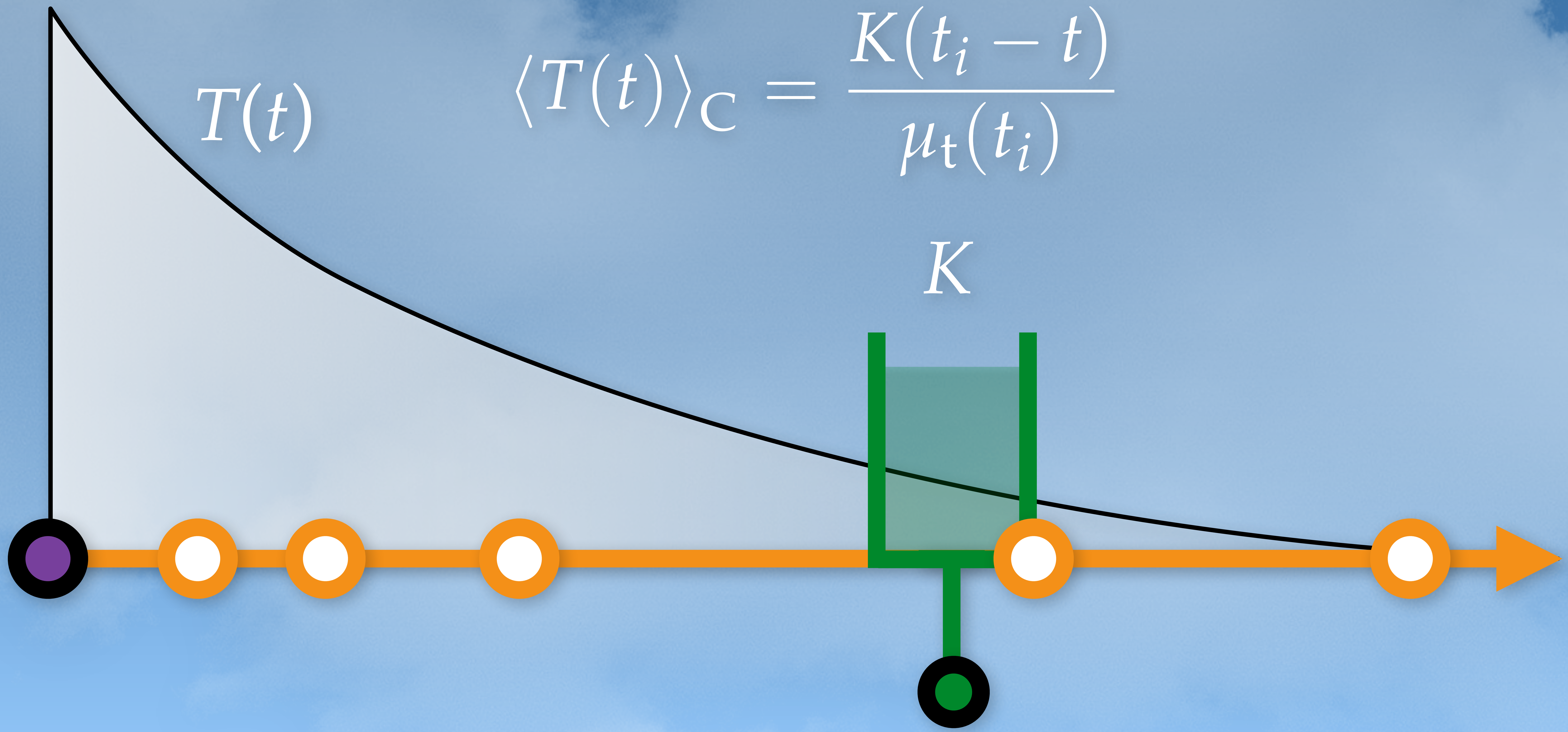
$$\langle T(t) \rangle_C = \frac{K(t_i - t)}{\mu_t(t_i)}$$



Counting free-flights

Collision estimator:

$$\langle T(t) \rangle_C = \frac{K(t_i - t)}{\mu_t(t_i)}$$



Why does this work?

Express transmittance as an integral (convolution with delta)

Why does this work?

Express transmittance as an integral (convolution with delta)

$$T(t) = \int_0^\infty T(s)\,\delta(s - t)\ \mathrm{d}s$$

Why does this work?

Express transmittance as an integral (convolution with delta)

Delta function

$$T(t) = \int_0^\infty T(s)\,\delta(s-t)\,\mathrm{d}s$$

Why does this work?

Express transmittance as an integral (convolution with delta)

Delta function

$$T(t) = \int_0^\infty T(s)\,\delta(s-t)\,\mathrm{d}s$$

Replace/blur into finite kernel

$$T(t) \approx \int_0^\infty T(s)\,K(s-t)\,\mathrm{d}s$$

Why does this work?

Express transmittance as an integral (convolution with delta)

Delta function

$$T(t) = \int_0^\infty T(s)\,\delta(s-t)\,\mathrm{d}s$$

Replace/blur into finite kernel

$$T(t) \approx \int_0^\infty T(s)\,K(s-t)\,\mathrm{d}s \approx \frac{T(t_i)\,K(t_i - t)}{p(t_i)}$$

Why does this work?

Express transmittance as an integral (convolution with delta)

$$T(t) = \int_0^\infty T(s)\,\delta(s-t)\,\mathrm{d}s$$

$$T(t) \approx \int_0^\infty T(s)\,K(s-t)\,\mathrm{d}s \approx \frac{T(t_i)\,K(t_i-t)}{p(t_i)}$$

Why does this work?

Express transmittance as an integral (convolution with delta)

$$T(t) = \int_0^\infty T(s)\,\delta(s-t)\,\mathrm{d}s$$

$$T(t) \approx \int_0^\infty T(s)\,K(s-t)\,\mathrm{d}s \approx \frac{T(t_i)\,K(t_i-t)}{p(t_i)}$$

$$\langle T(t)\rangle_C = \frac{T(t_i)\,K(|t_i-t|)}{\mu_{\mathrm{t}}(t_i)T(t_i)}$$

Why does this work?

Express transmittance as an integral (convolution with delta)

$$T(t) = \int_0^\infty T(s)\,\delta(s-t)\;\mathrm{ds}$$

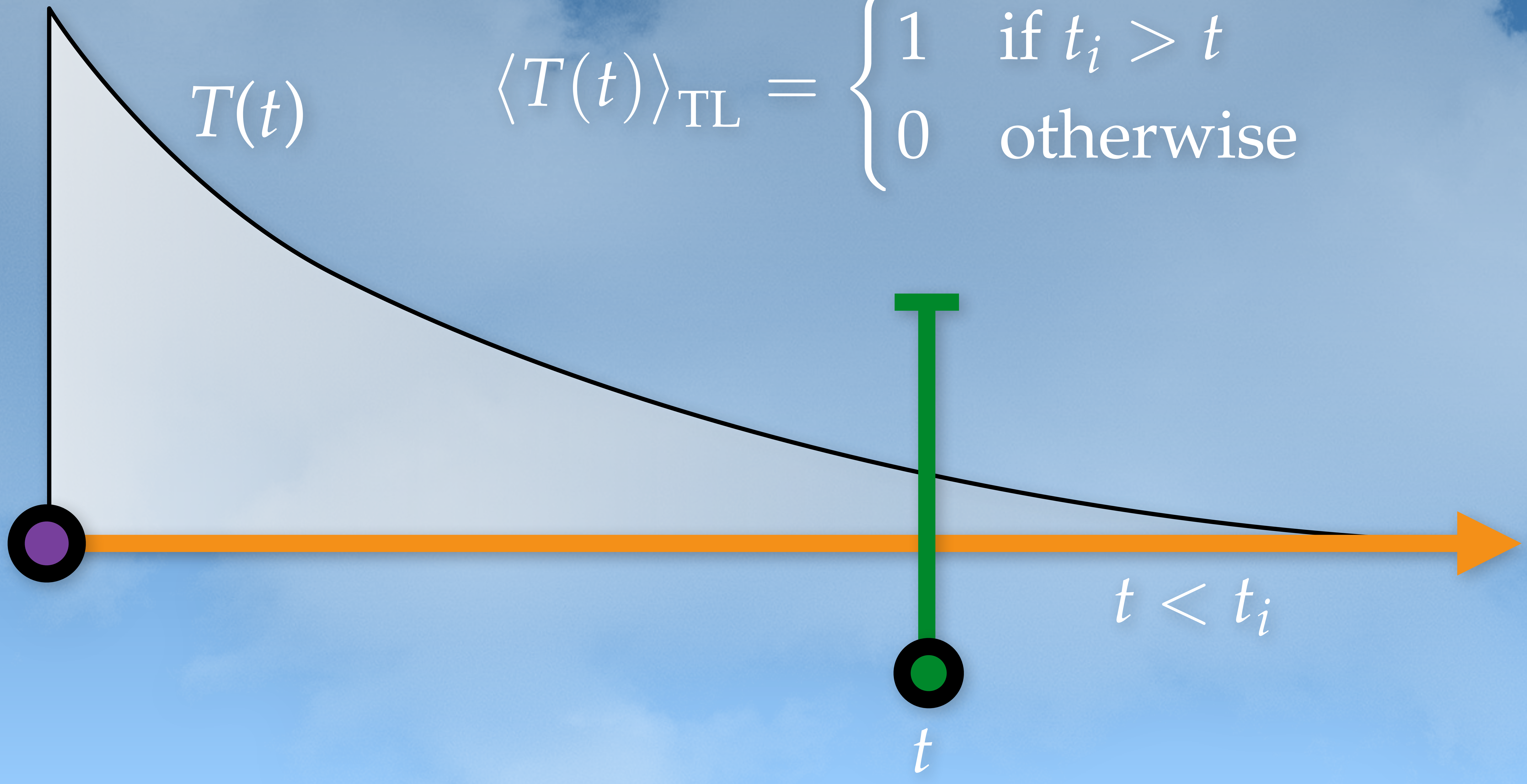
$$T(t) \approx \int_0^\infty T(s)\,K(s-t)\;\mathrm{ds} \approx \frac{T(t_i)\,K(t_i-t)}{p(t_i)}$$

$$\langle T(t)\rangle_C = \frac{\cancel{T(t_i)}\,K(|t_i-t|)}{\mu_\mathrm{t}(t_i)\cancel{T(t_i)}} = \frac{K(t_i-t)}{\mu_\mathrm{t}(t_i)}$$

Counting free-flights

Track-length estimator (binary):

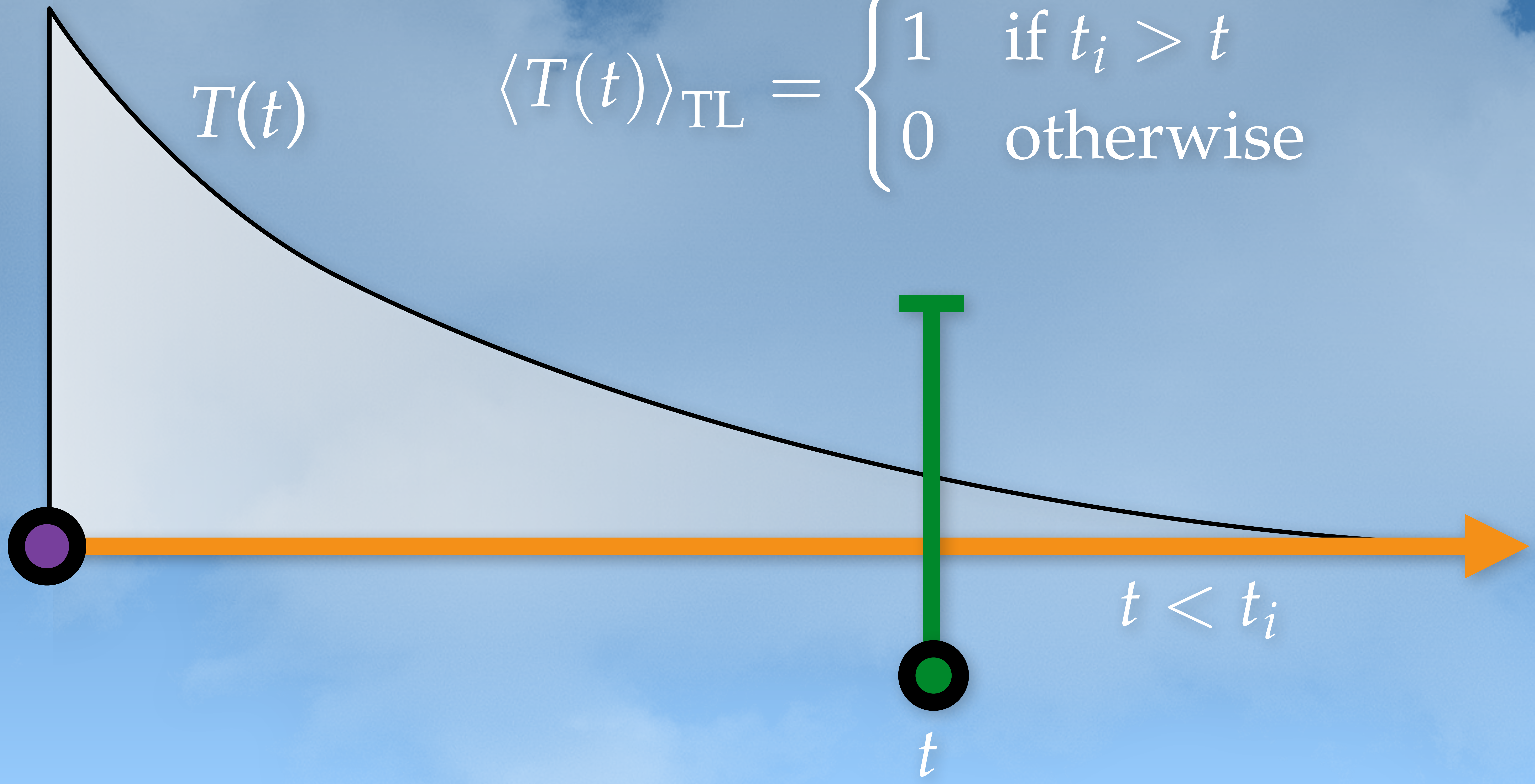
$$\langle T(t) \rangle_{\mathrm{TL}} = \begin{cases} 1 & \text{if } t_i > t \\ 0 & \text{otherwise} \end{cases}$$



Counting free-flights

Track-length estimator (binary):

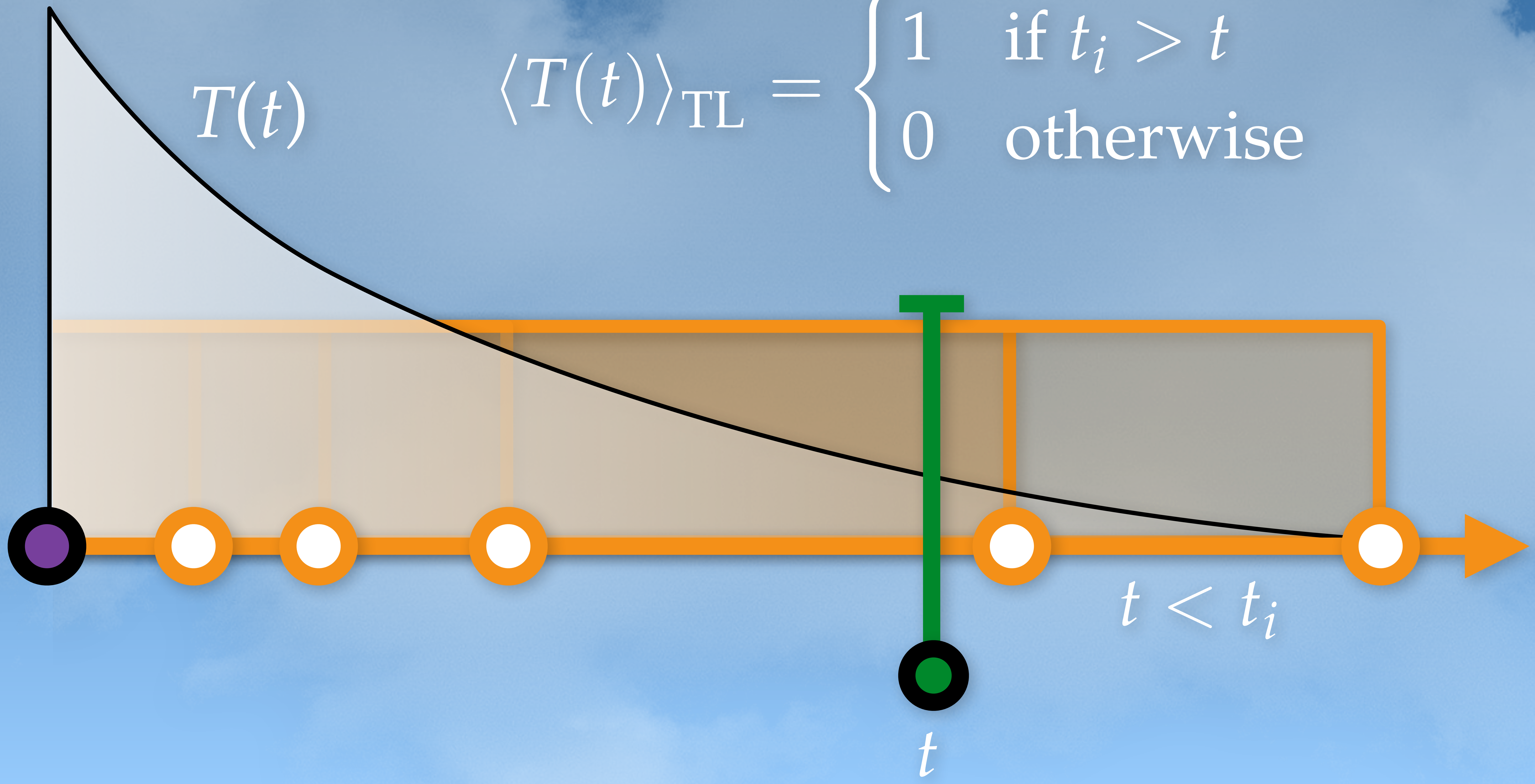
$$\langle T(t) \rangle_{\mathrm{TL}} = \begin{cases} 1 & \text{if } t_i > t \\ 0 & \text{otherwise} \end{cases}$$



Counting free-flights

Track-length estimator (binary):

$$\langle T(t)\rangle_{\mathrm{TL}} = \begin{cases} 1 & \text{if } t_i > t \\ 0 & \text{otherwise} \end{cases}$$



Why does this work?

Why does this work?

Start with Russian roulette (unbiased):

$$\langle T(t) \rangle_{\mathrm{RR}} = \begin{cases} \frac{T(t)}{P(accept)} & \text{if } accept \\ 0 & \text{otherwise.} \end{cases}$$

Why does this work?

Start with Russian roulette (unbiased):

$$\langle T(t) \rangle_{\mathrm{RR}} = \begin{cases} \frac{T(t)}{P(accept)} & \text{if } accept \\ 0 & \text{otherwise.} \end{cases}$$

Using free-flight sampling for acceptance probability:

$$P(t_i > t) = \int_t^\infty p(s)\, \mathrm{d}s$$

Why does this work?

Start with Russian roulette (unbiased):

$$\langle T(t) \rangle_{\mathrm{RR}} = \begin{cases} \frac{T(t)}{P(accept)} & \text{if } accept \\ 0 & \text{otherwise.} \end{cases}$$

Using free-flight sampling for acceptance probability:

$$P(t_i > t) = \int_t^\infty p(s)\, \mathrm{d}s = \int_t^\infty \mu_{\mathrm{t}}(s)\, T(s)\, \mathrm{d}s = T(t)$$

Why does this work?

Start with Russian roulette (unbiased):

$$\langle T(t) \rangle_{\mathrm{RR}} = \begin{cases} \frac{T(t)}{P(accept)} & \text{if } accept \\ 0 & \text{otherwise.} \end{cases}$$

Using free-flight sampling for acceptance probability:

$$P(t_i > t) = \int_t^\infty p(s)\,\mathrm{d}s = \int_t^\infty \mu_{\mathrm{t}}(s)\,T(s)\,\mathrm{d}s = T(t)$$

This gives an **unbiased** "track-length estimator":

$$\langle T(t) \rangle_{\mathrm{TL}} = \begin{cases} \frac{T(t)}{P(t_i > t)} = 1 & \text{if } t_i > t \\ 0 & \text{otherwise} \end{cases}$$

Transmittance from free-flight sampling

Transmittance from free-flight sampling

- ✔ Estimate transmittance given free-flight sampling

Transmittance from free-flight sampling

✔ Estimate transmittance given free-flight sampling

Collision Estimator

- Count photons starting at x that land *near* y

Transmittance from free-flight sampling

✔ Estimate transmittance given free-flight sampling

Collision Estimator

- Count photons starting at x that land *near* y

✘ Biased (transmittance is blurred due to kernel/bin size)

Transmittance from free-flight sampling

✔ Estimate transmittance given free-flight sampling

Collision Estimator

- Count photons starting at x that land *near* y

✘ Biased (transmittance is blurred due to kernel/bin size)

Track-length Estimator

- Count photons starting at x that travel *past* y

Transmittance from free-flight sampling

✔ Estimate transmittance given free-flight sampling

Collision Estimator

- Count photons starting at x that land *near* y

✘ Biased (transmittance is blurred due to kernel/bin size)

Track-length Estimator

- Count photons starting at x that travel *past* y

✔ Unbiased!

Transmittance estimation

1. Estimators integrating **optical thickness**
2. Estimators using **free-flight sampling**
3. **Next:** Estimators using **null collisions**

$$T(\mathbf{x}, \mathbf{y}) = \mathrm{e}^{-\tau(\mathbf{x}, \mathbf{y})}$$

transmittance

$$\tau(\mathbf{x}, \mathbf{y}) = \int_0^t \mu_{\mathrm{t}}(\mathbf{x} - s\omega)\,\mathrm{d}s$$

optical thickness

$$p(t_i) \propto T(t_i)$$

free-flight sampling