Analytical Motion Blur Rasterization with Compression

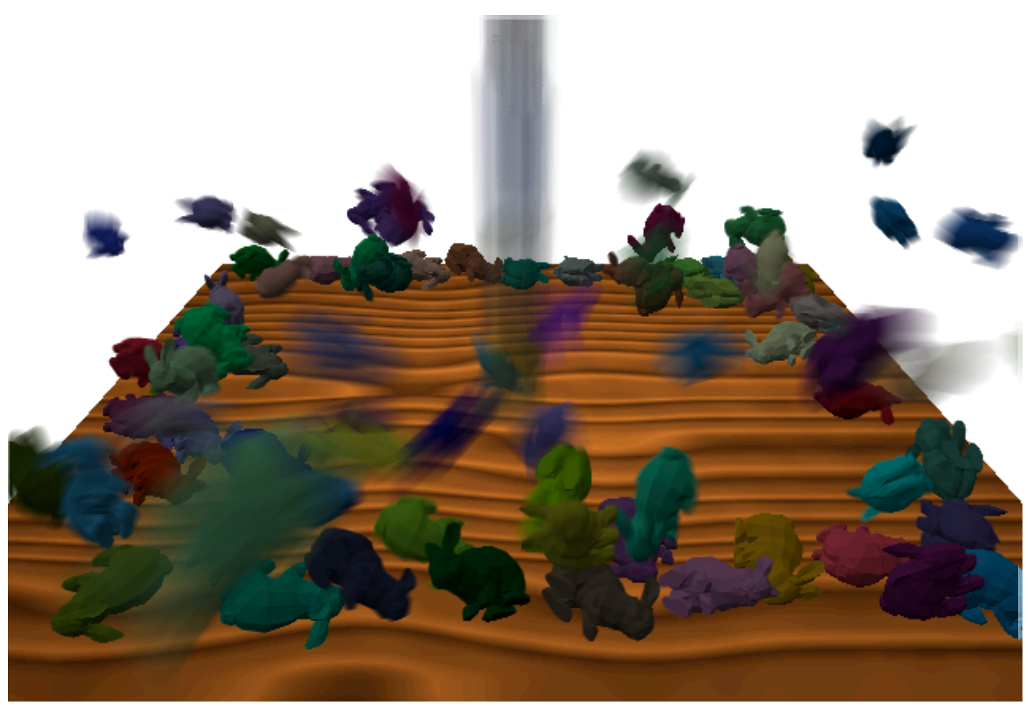
Carl Johan Gribel[1]
Michael Doggett[1]
Tomas Akenine-Möller[1,2]

[1]Lund University
[2]Intel Corproration

Motivation

Motivation

- Human visual system designed to detect motion
- This fails when presented too few images, or too much motion per image
  - motion gets jumpy
- Motion blur aids the motion detection of the visual system



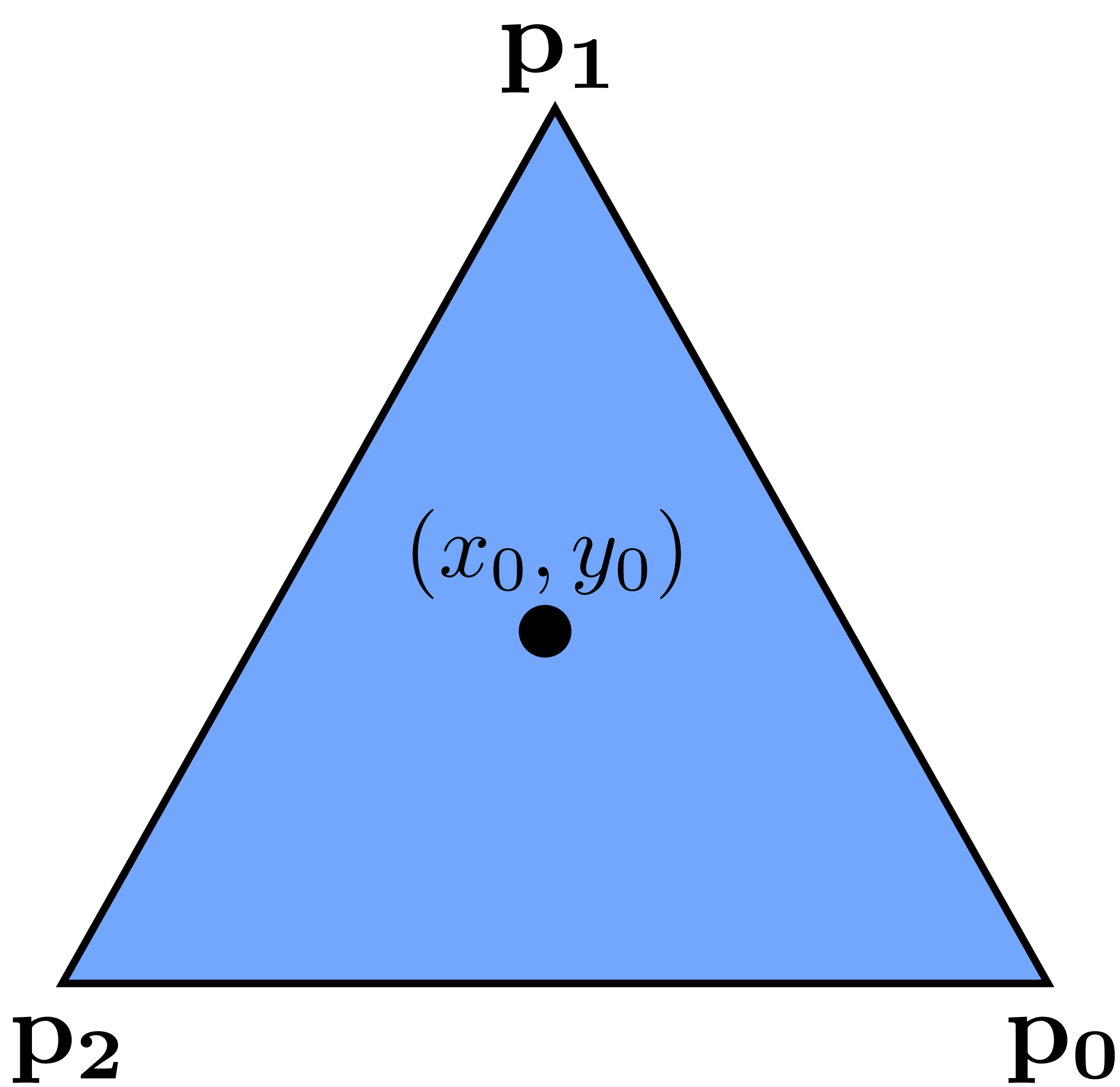
Talk outline

- Edge Equations and exact exposure intervals
  - analytic inside-test
  - visibility management
- Computing the time integral
  - opaque/translucent resolve
- Compression

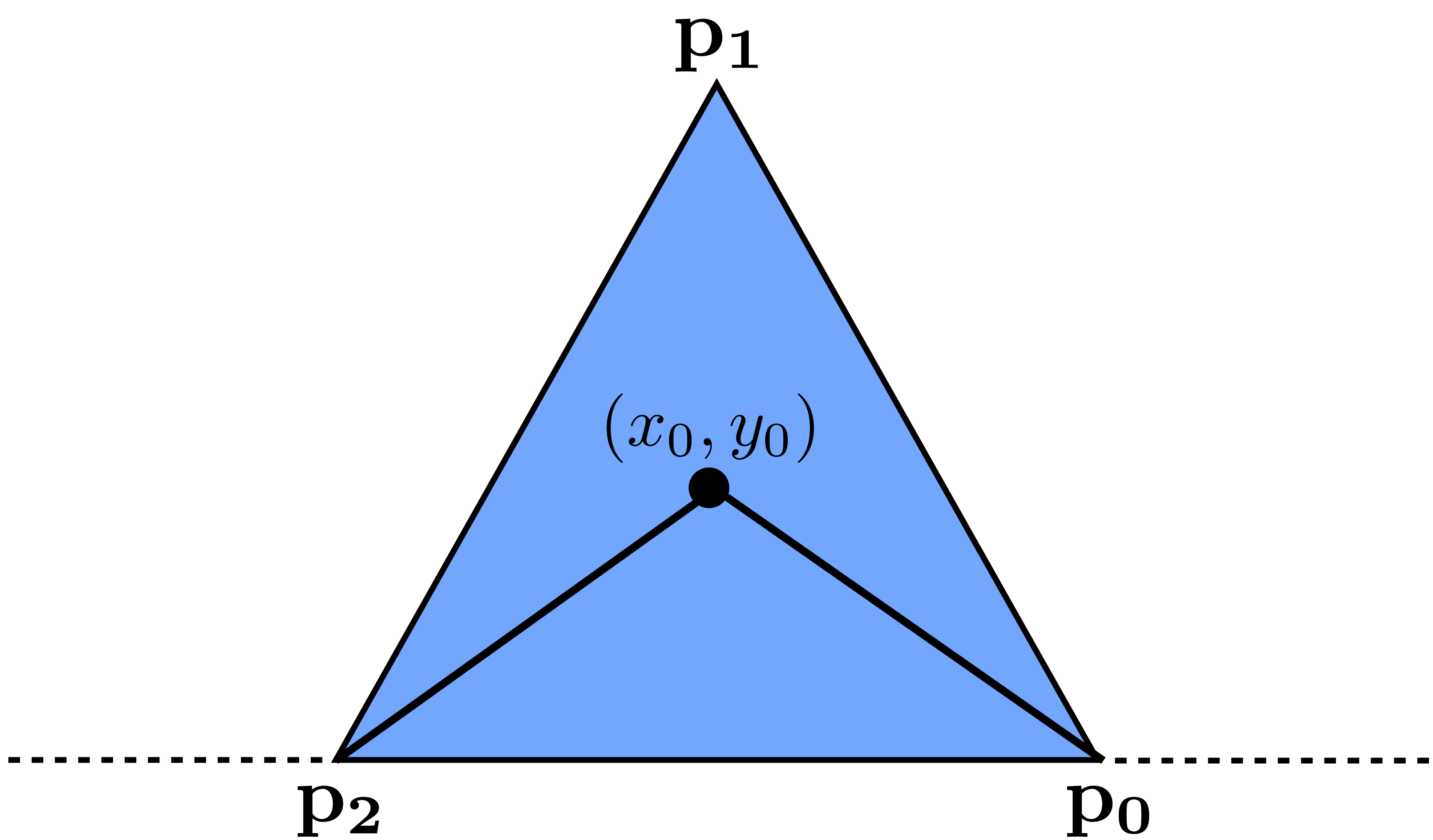
Previous work

- Accumulation buffer [HA90][WGER05][DWS88]
  - expensive
  - strobing artifacts
- Stochastic [CCC87] [AMMH07][McGuire et al. HPG2010]
  - correct solution but slow convergence
- Single sample analytical visibility [KB83]
  Extended with stochastic shading by Sung et al. [SPW02]
  - details missing for visibility computations

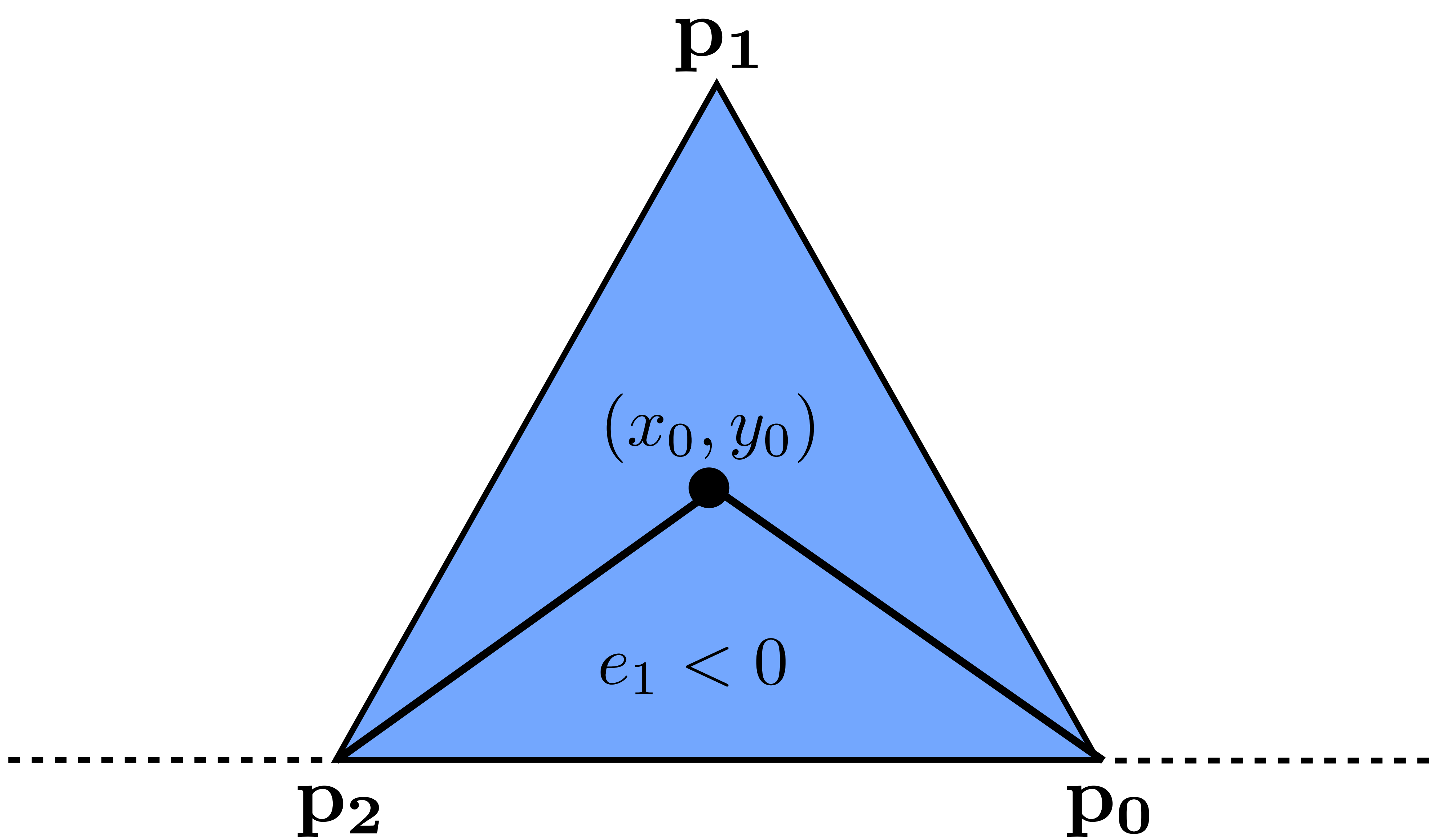
Edge Equation primer for static triangle



Edge Equation primer for static triangle

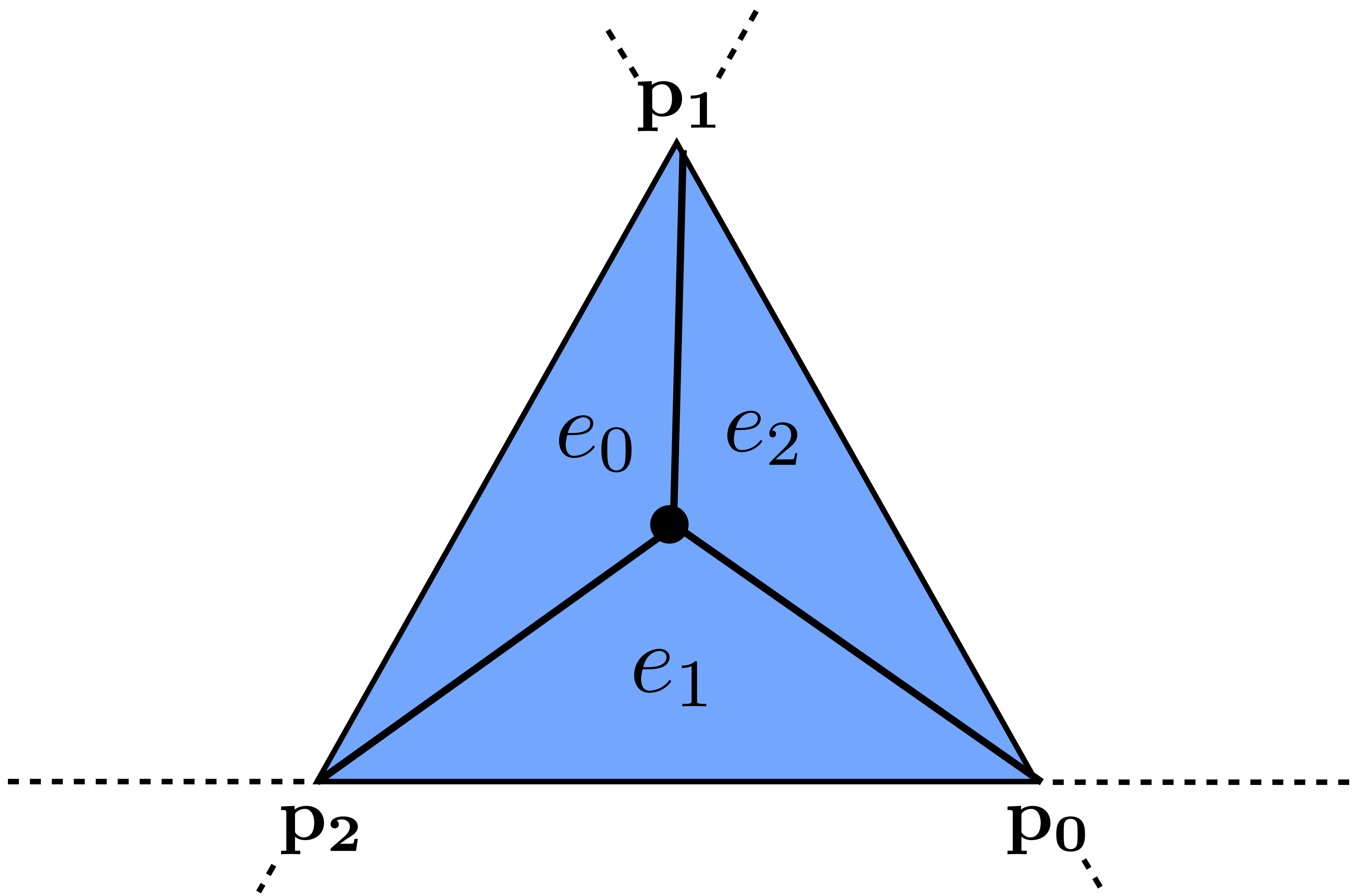


Edge Equation primer for static triangle



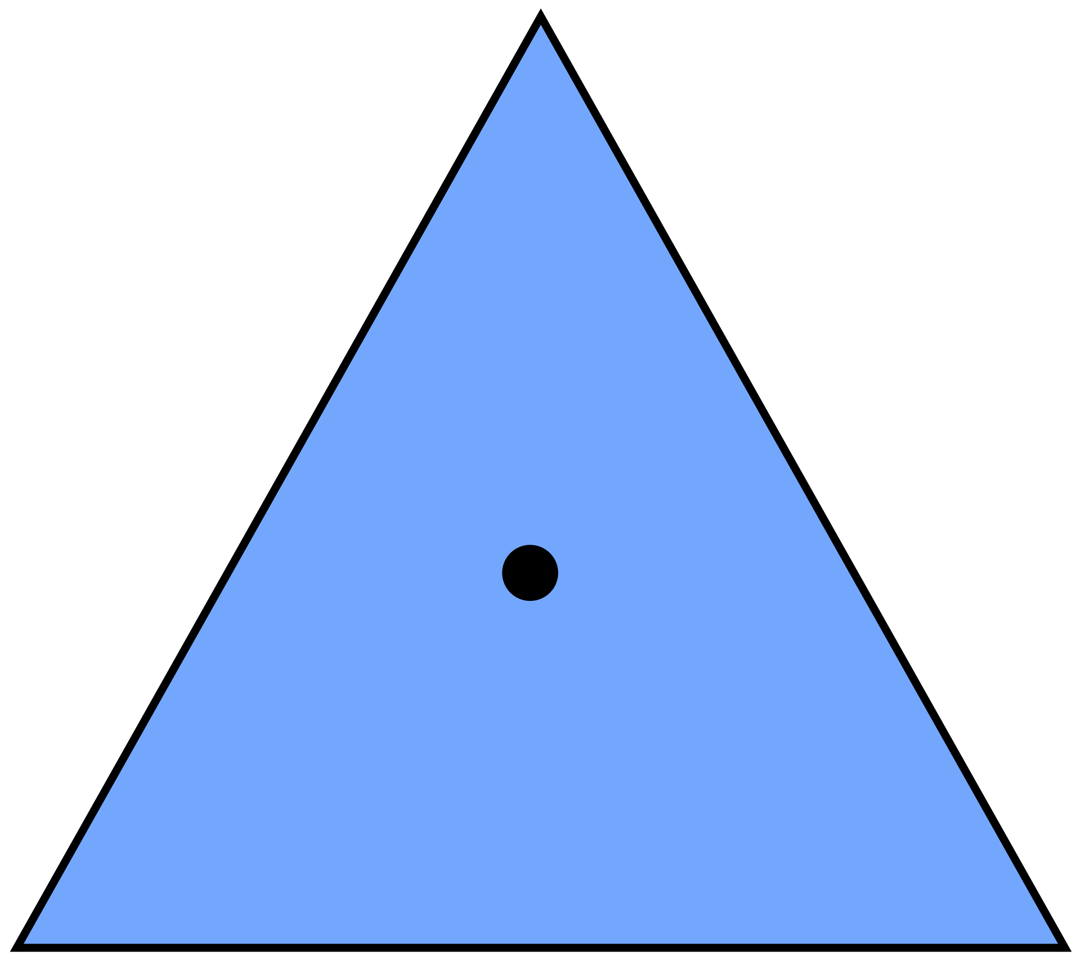
3D homogeneous space: $e_1 = (\mathbf{p_2} \times \mathbf{p_0}) \cdot (x_0, y_0, 1)$

Edge Equation primer for static triangle

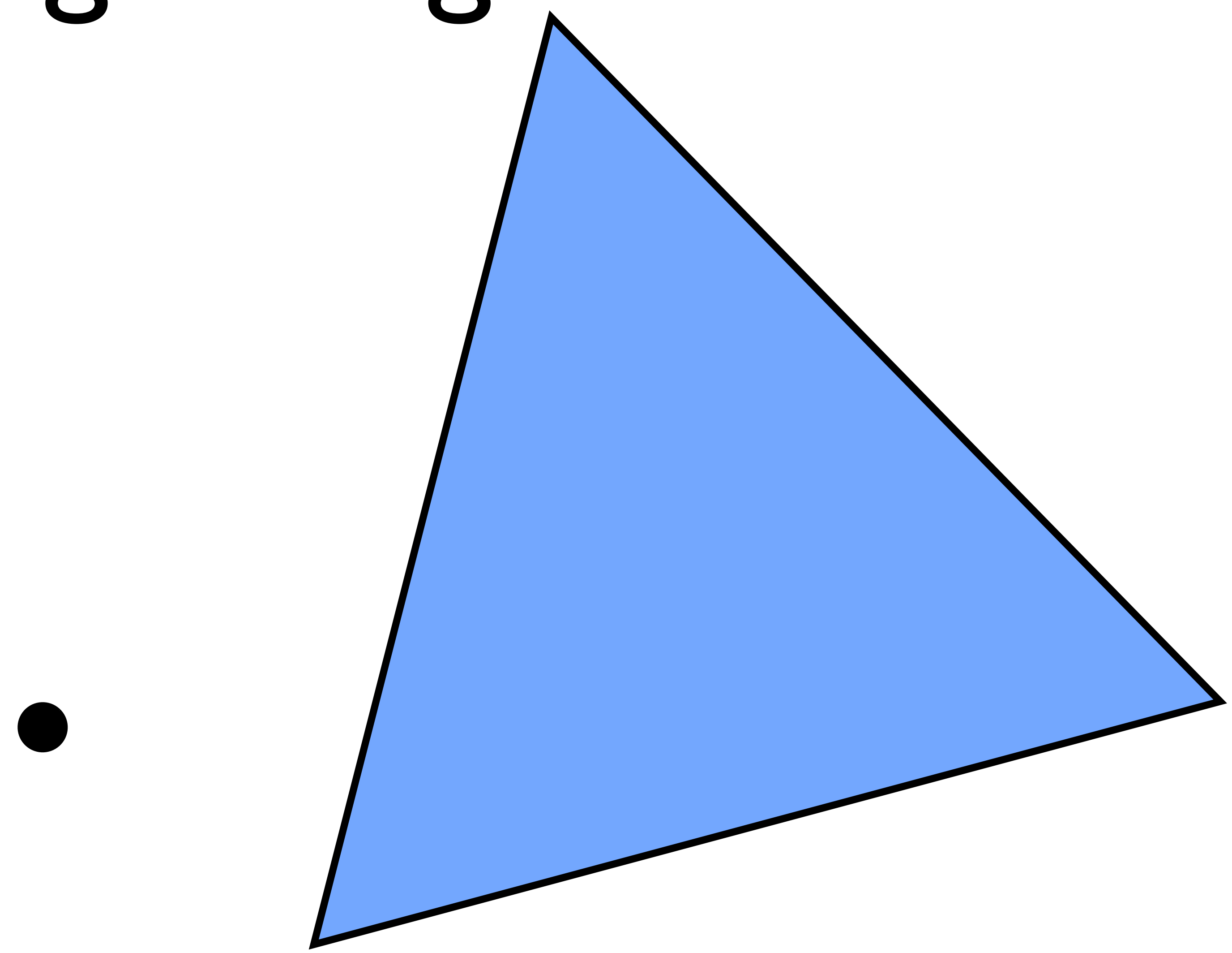


3D homogeneous space: $e_1 = (\mathbf{p_2} \times \mathbf{p_0}) \cdot (x_0, y_0, 1)$

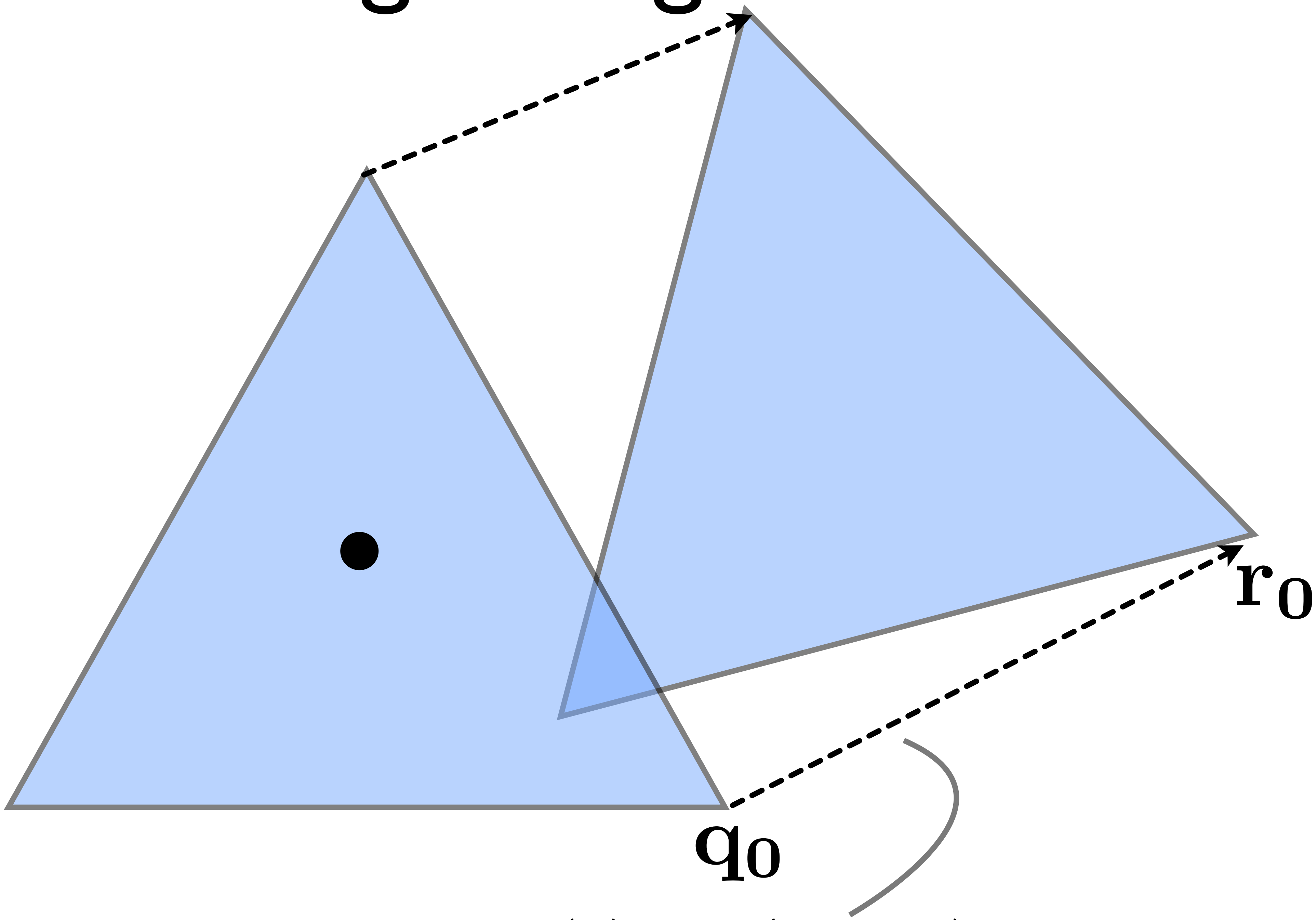
Edge Equation primer for moving triangle



Edge Equation primer for moving triangle



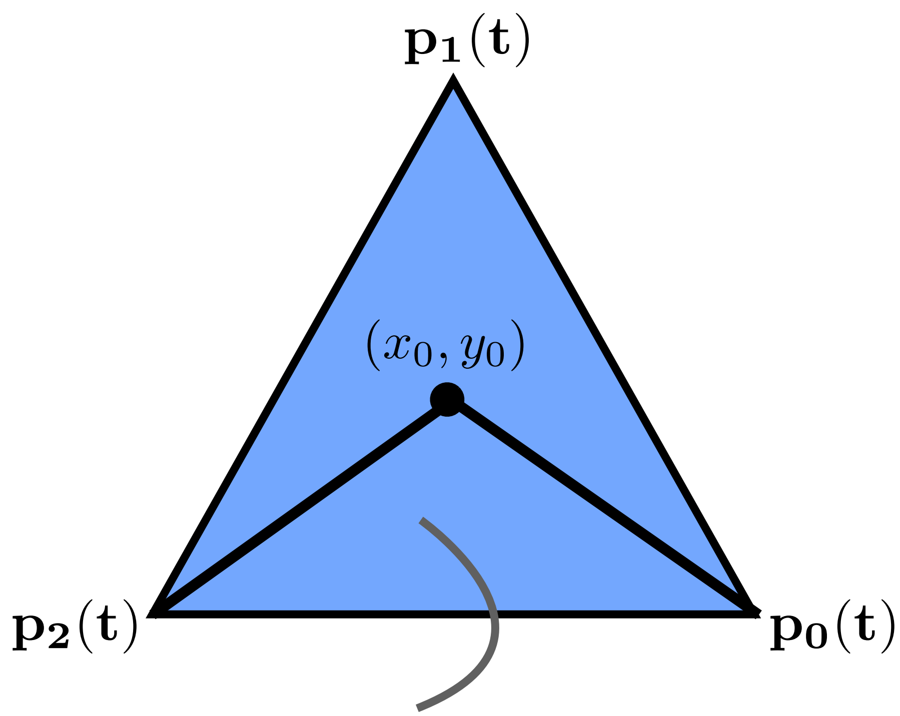
Edge Equation primer for moving triangle



$$\mathbf{p_0}(t) = (1-t)\mathbf{q_0} + t\mathbf{r_0}$$

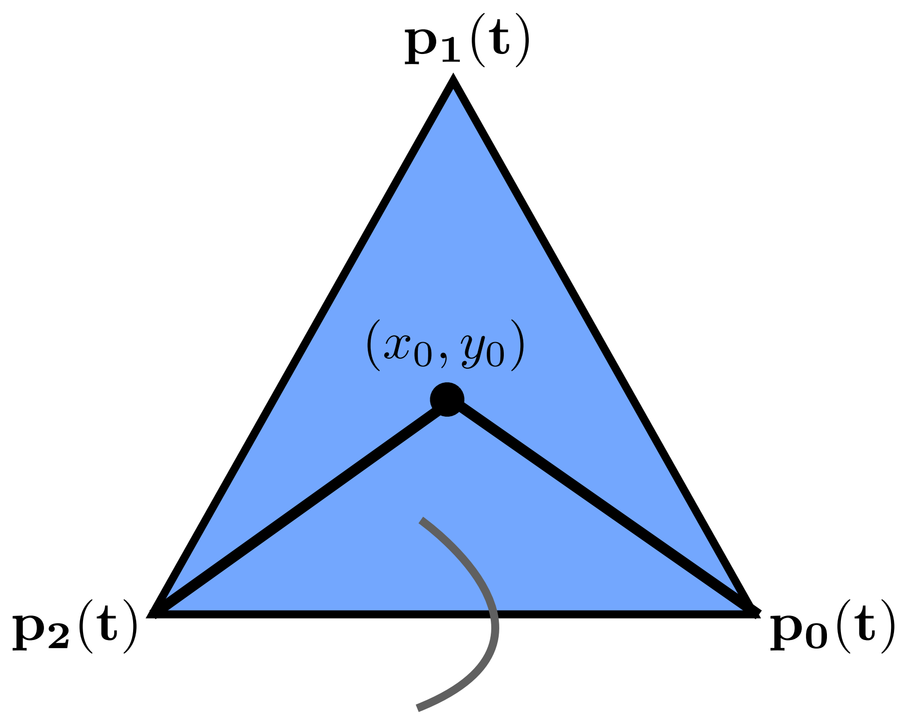
$$t \in [0,1]$$

Edge Equations



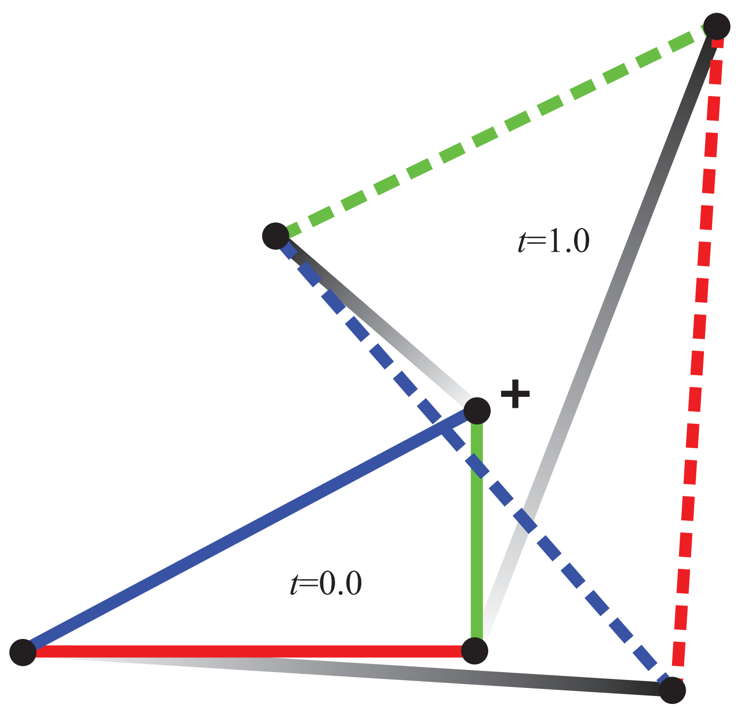
$$e_1(t) = (\mathbf{p_2}(\mathbf{t}) \times \mathbf{p_0}(\mathbf{t})) \cdot (x_0, y_0, 1)$$
$$= (((1-t)\mathbf{q_2} + t\mathbf{r_2}) \times ((1-t)\mathbf{q_0} + t\mathbf{r_0})) \cdot (x_0, y_0, 1)$$
$$= \boxed{(\mathbf{f}t^2 + \mathbf{g}t + \mathbf{h}) \cdot (x_0, y_0, 1)}$$

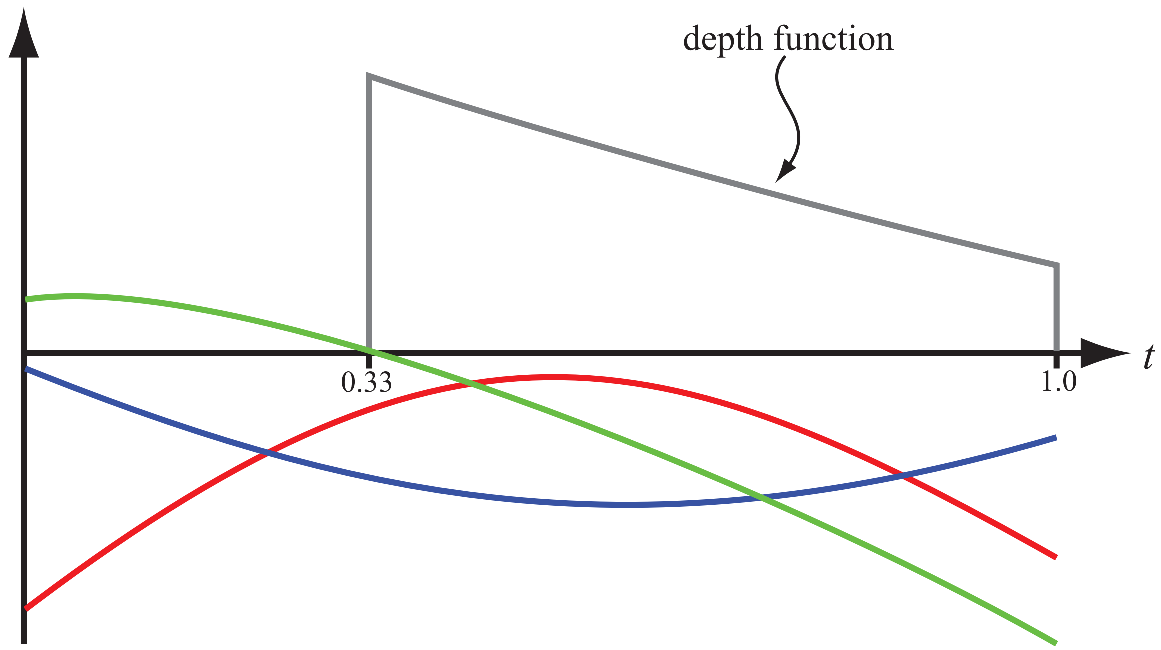
Edge Equations

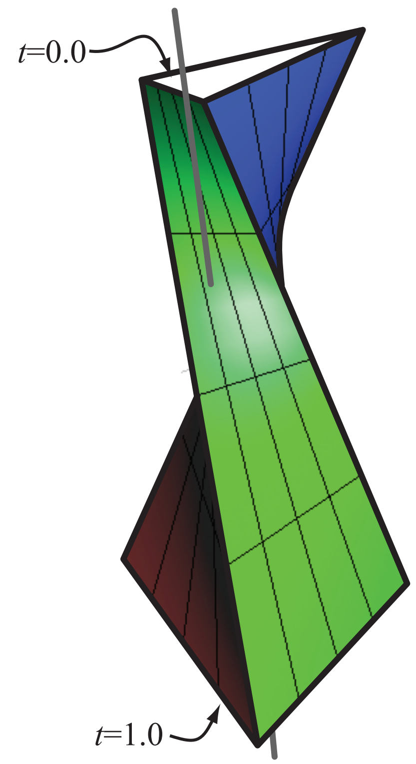


$$\begin{aligned} e_1(t) &= (\mathbf{p_2}(\mathbf{t}) \times \mathbf{p_0}(\mathbf{t})) \cdot (x_0, y_0, 1) \\ &= (((1-t)\mathbf{q_2} + t\mathbf{r_2}) \times ((1-t)\mathbf{q_0} + t\mathbf{r_0})) \cdot (x_0, y_0, 1) \\ &= \boxed{(\mathbf{f}t^2 + \mathbf{g}t + \mathbf{h}) \cdot (x_0, y_0, 1)} \end{aligned}$$

We also derive the analytical depth function $d(t)$ ...

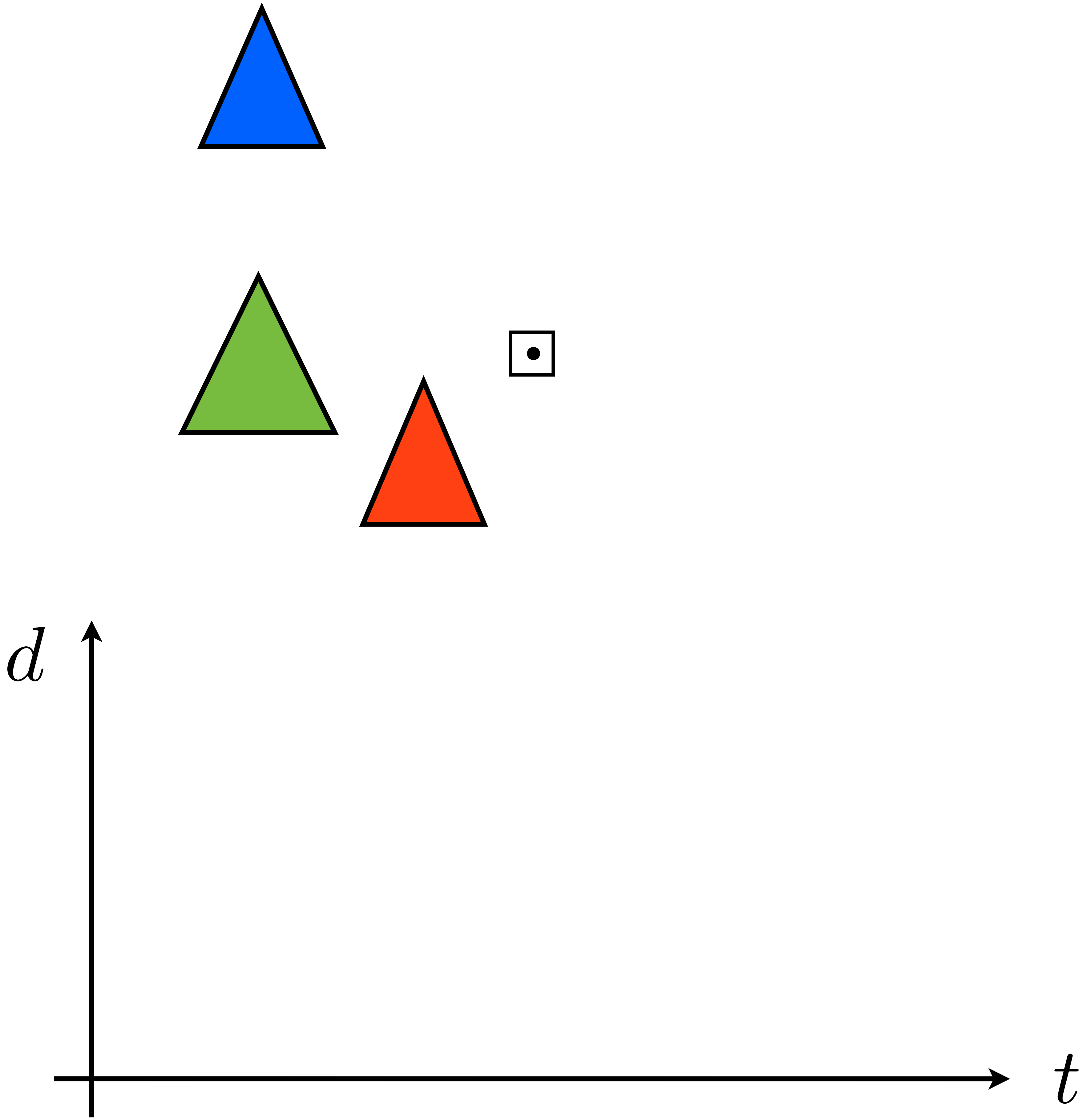

t=1.0
+
t=0.0


depth function
0.33
1.0
t

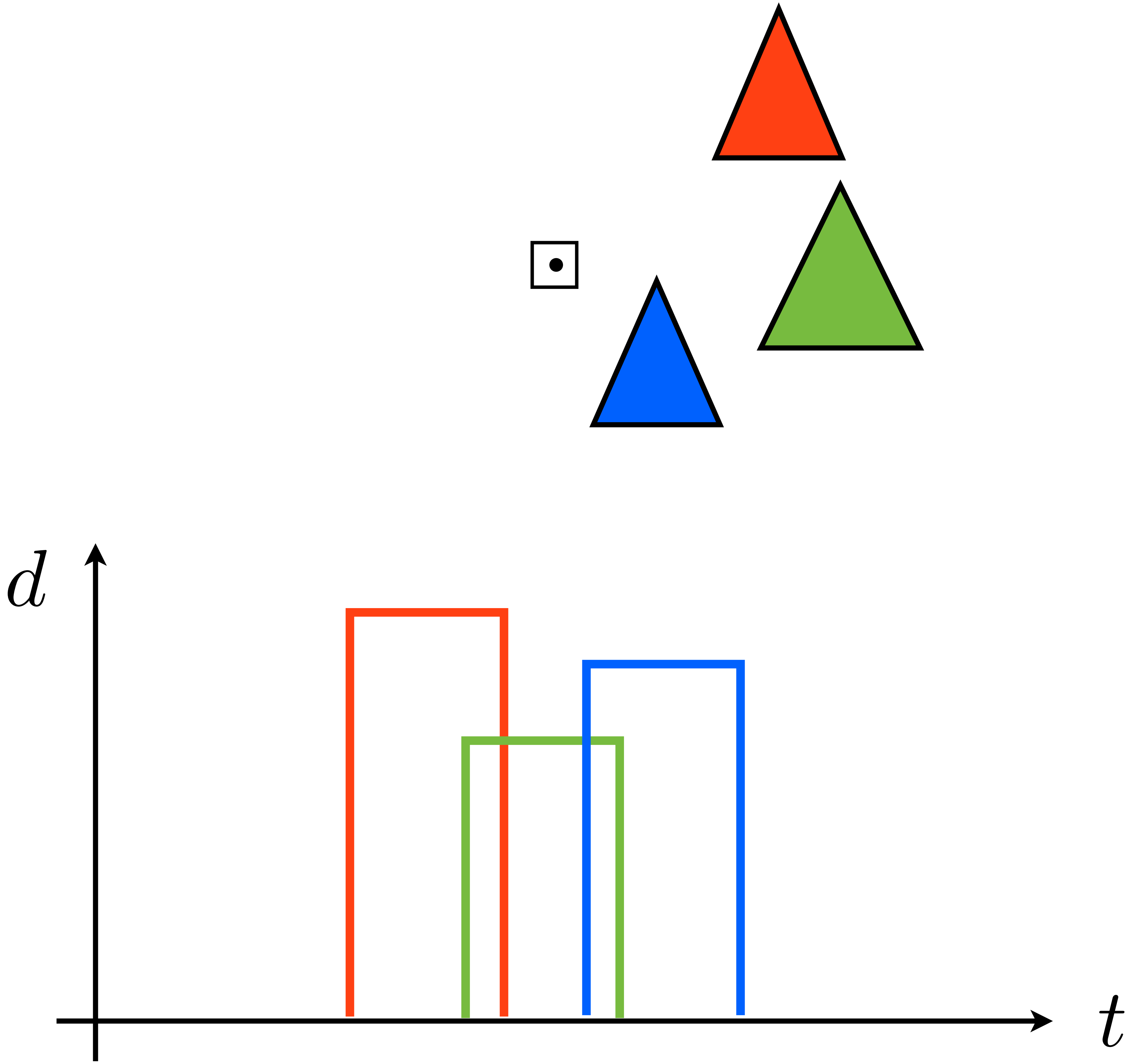

t=0.0
t=1.0

Rasterization

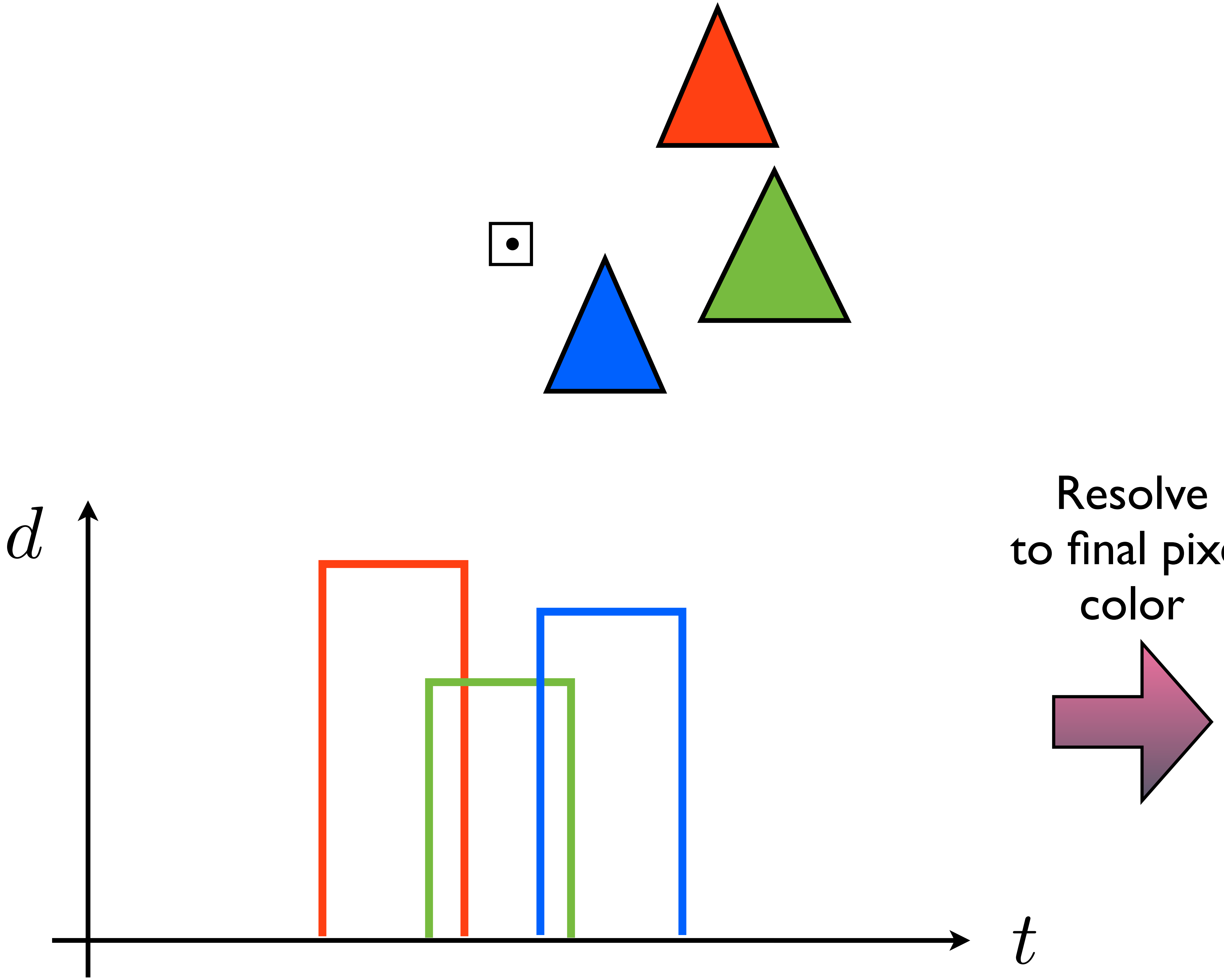
- For each pixel of bounding box, compute 2nd degree solutions
- Insert intervals into per-pixel list



d
t

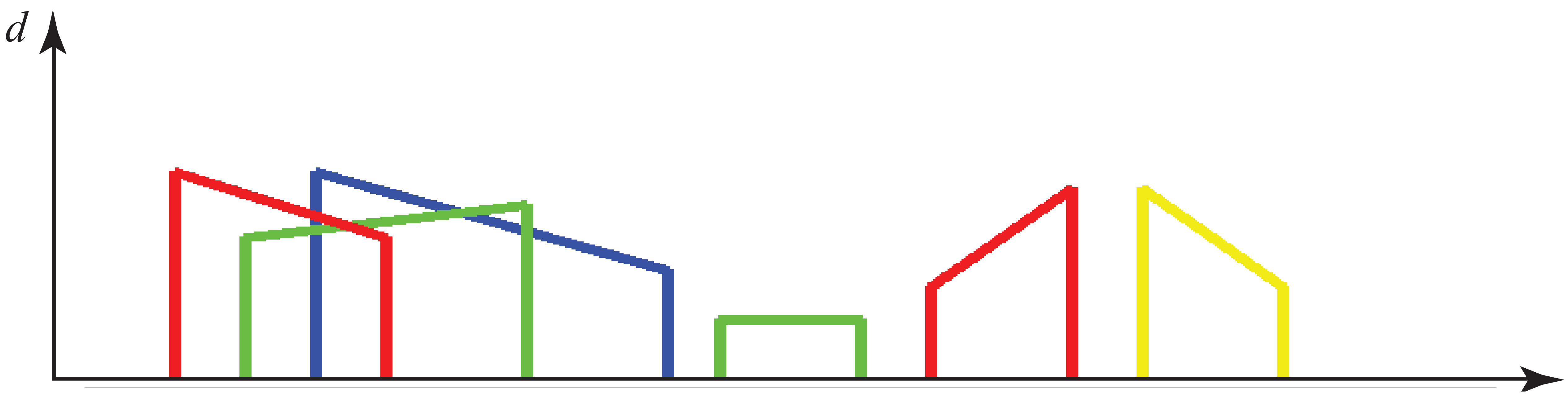


d
t


d
t
Resolve
to final pixel
color

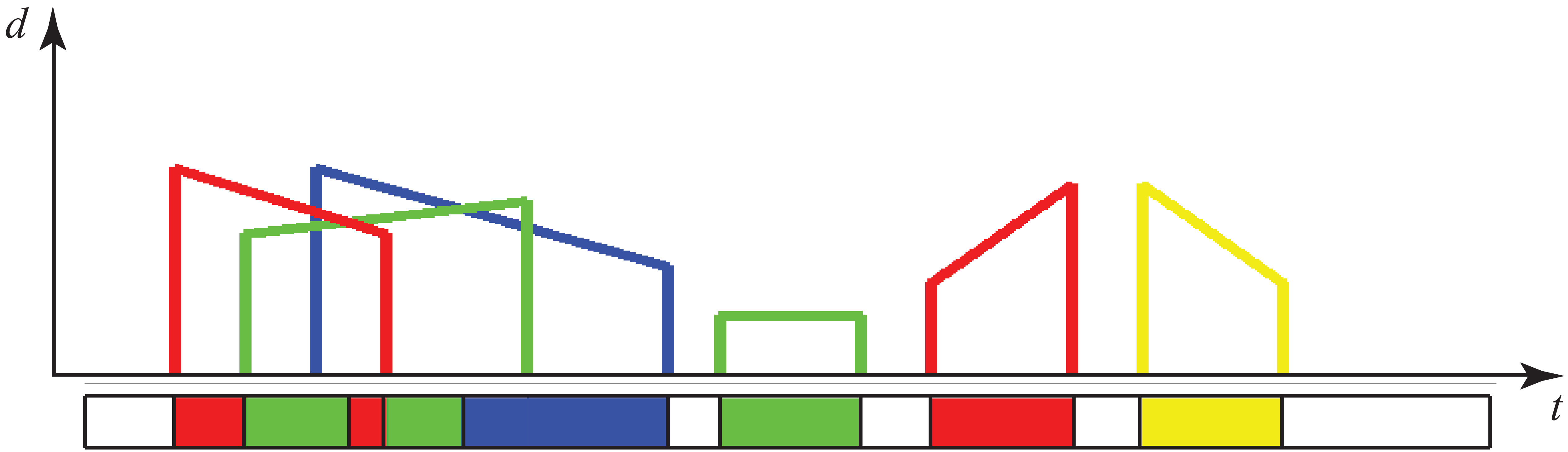
Opaque resolve

- Sweep over time and keep track of overlapping intervals
- Pick colors of the closest intervals
- Blend together for final color



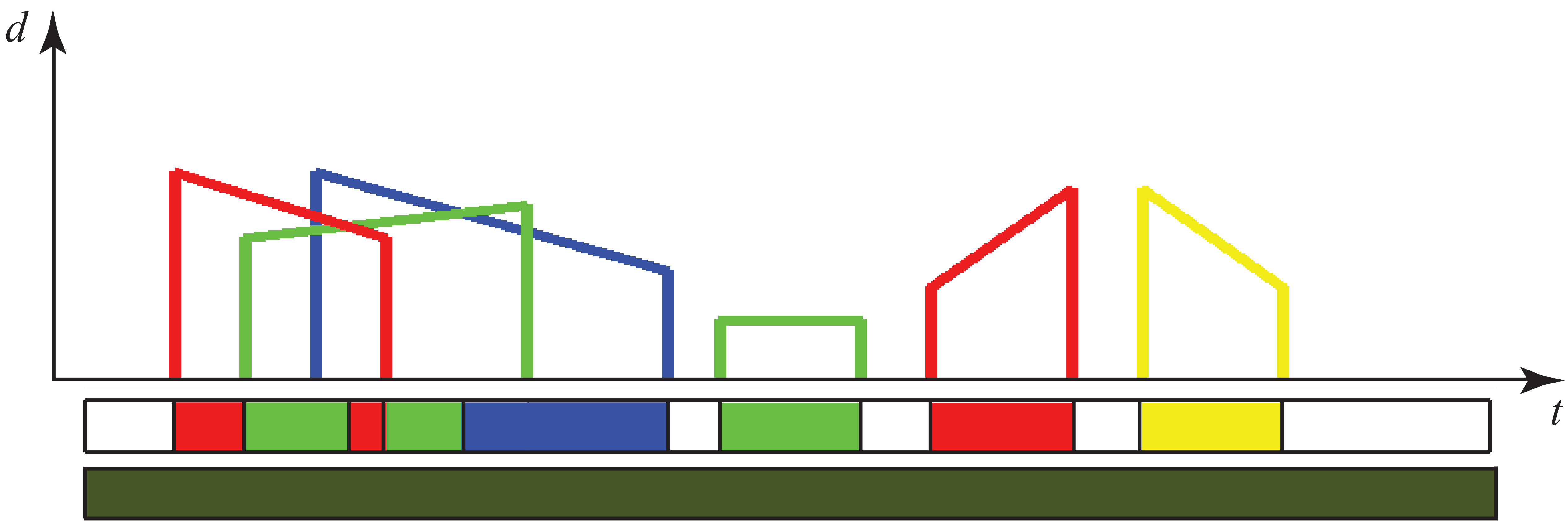
Opaque resolve

- Sweep over time and keep track of overlapping intervals
- Pick colors of the closest intervals
- Blend together for final color



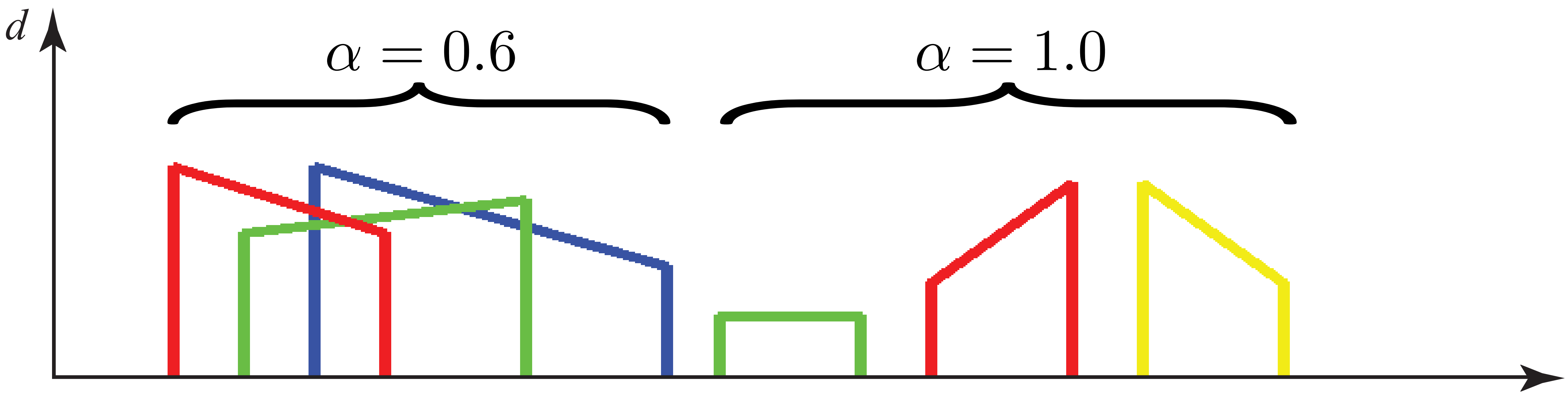
Opaque resolve

- Sweep over time and keep track of overlapping intervals
- Pick colors of the closest intervals
- Blend together for final color



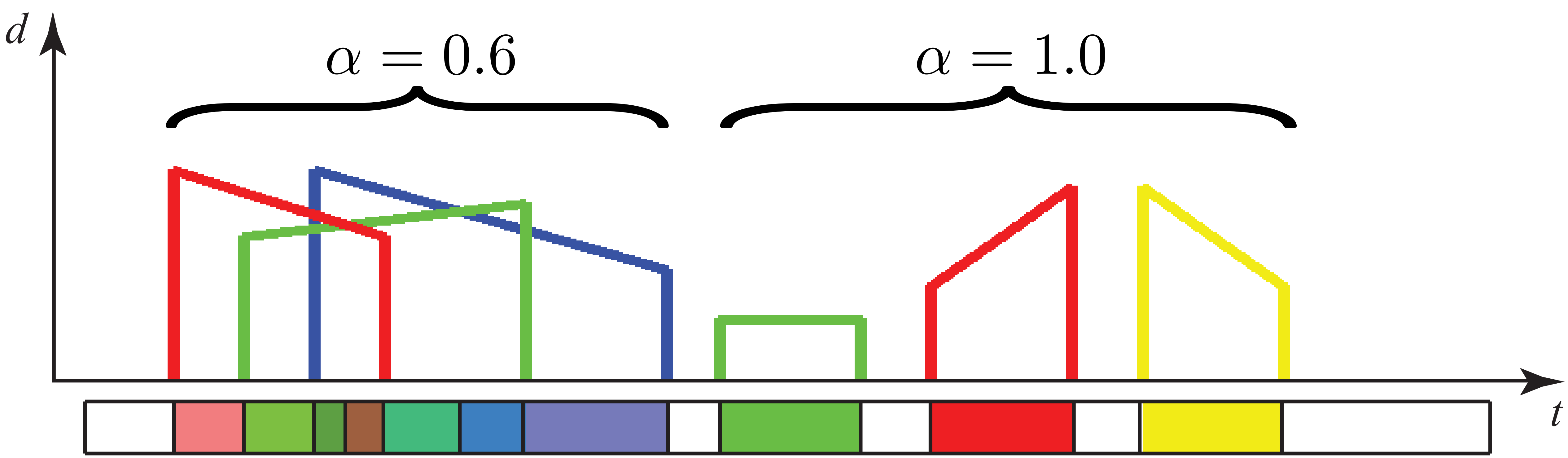
Translucent resolve

- Sweep over time and keep track of overlapping intervals
- **Alpha-blend each subsection**
- Blend together for final color



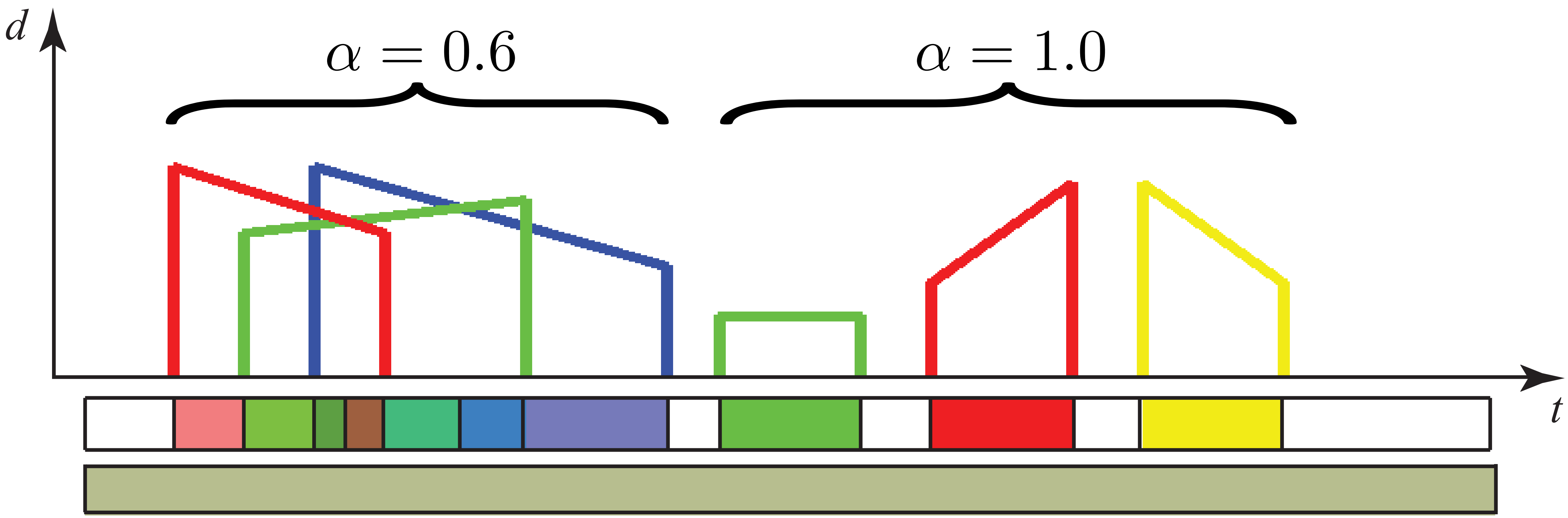
Translucent resolve

- Sweep over time and keep track of overlapping intervals
- **Alpha-blend each subsection**
- Blend together for final color



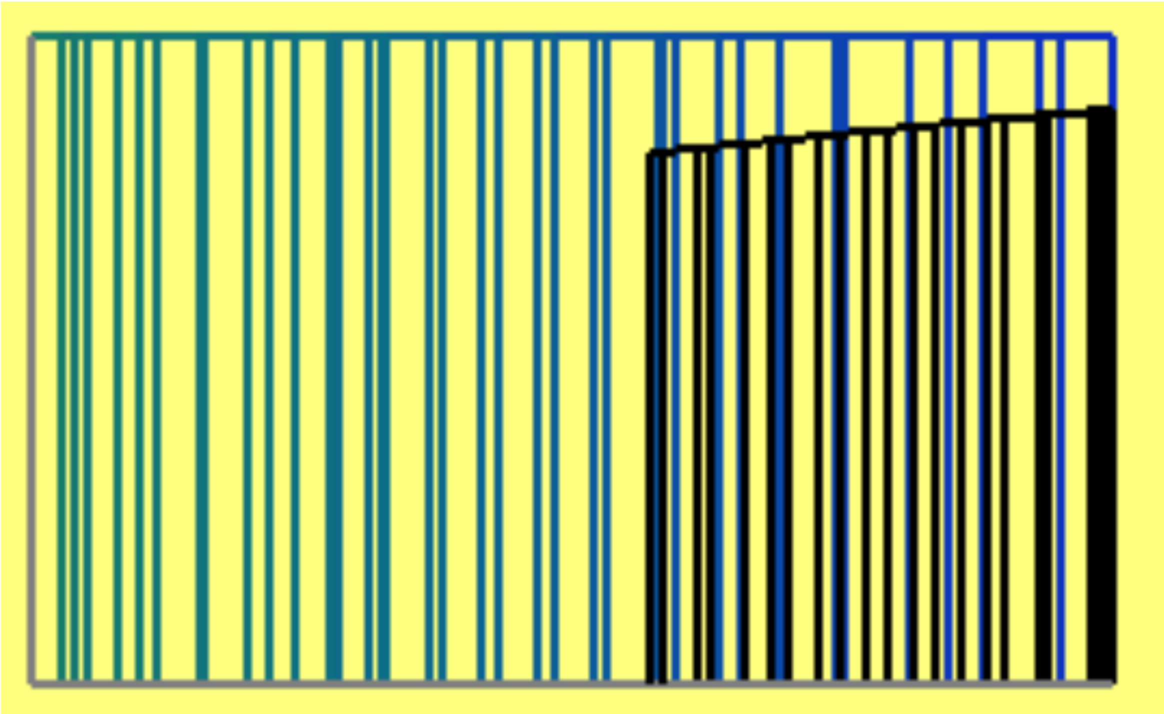
Translucent resolve

- Sweep over time and keep track of overlapping intervals
- **Alpha-blend each subsection**
- Blend together for final color



Compression

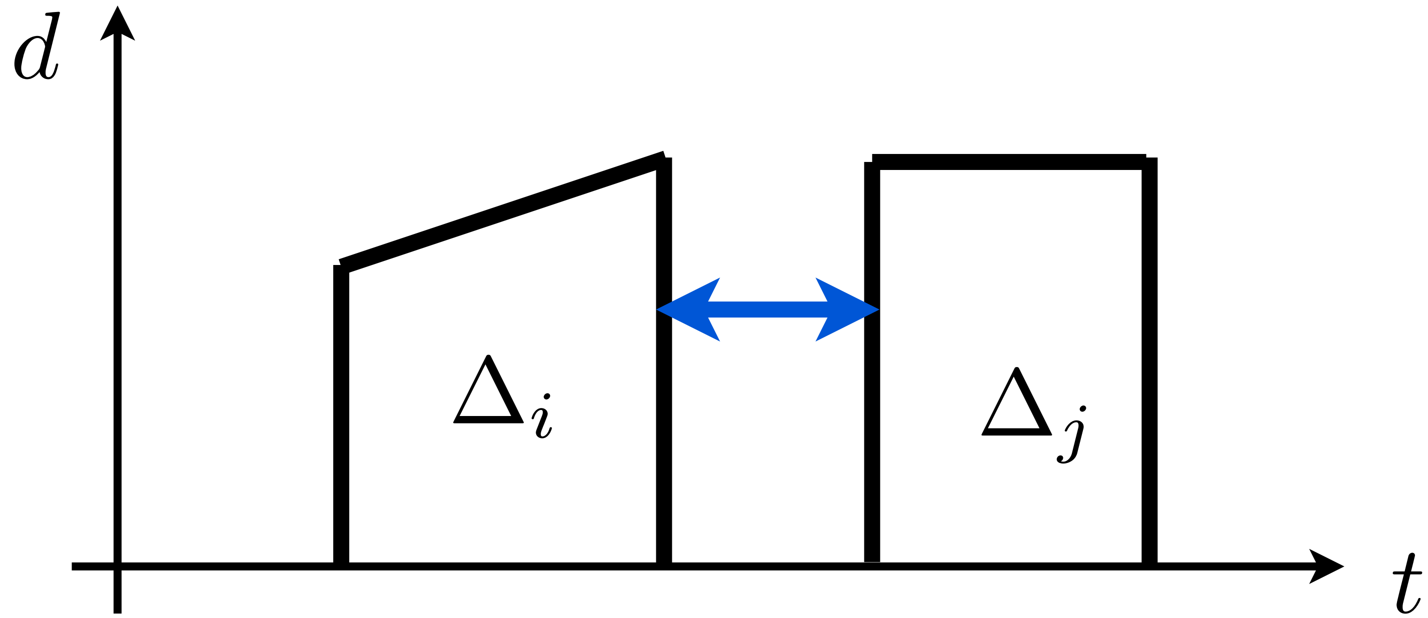
- Will end up with lots of intervals
- But, can expect spatial & temporal coherency
- Proposal: merge similar intervals



Compression: Oracle function

$$O(\Delta_i, \Delta_j)$$

- Metric for interval similarity
- Merge pairs of intervals deemed most similar



$$O(\Delta_i, \Delta_j) = h_1 \max(t_j^s - t_i^e, 0) +$$

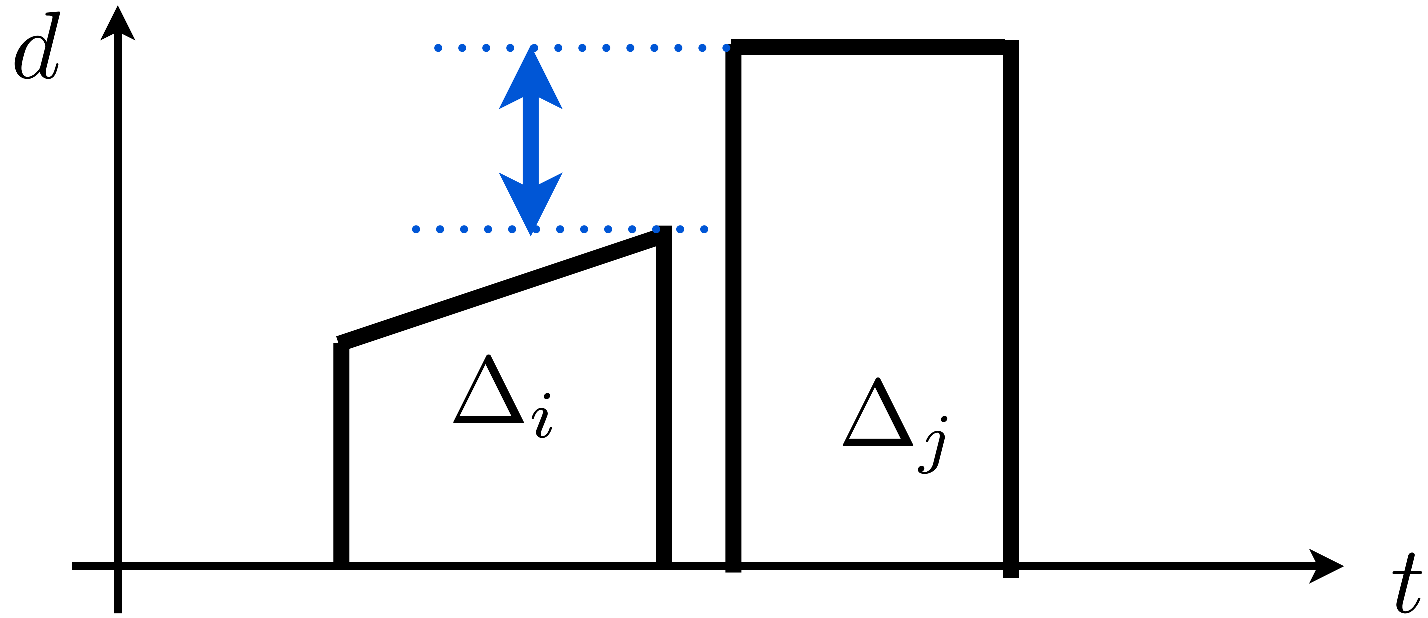
temporal proximity

$$h_2|\bar{z}_i - z_j| +$$

$$h_3|k_i - k_j| +$$

$$h_4(t_i^e - t_i^s + t_j^e - t_j^s) +$$

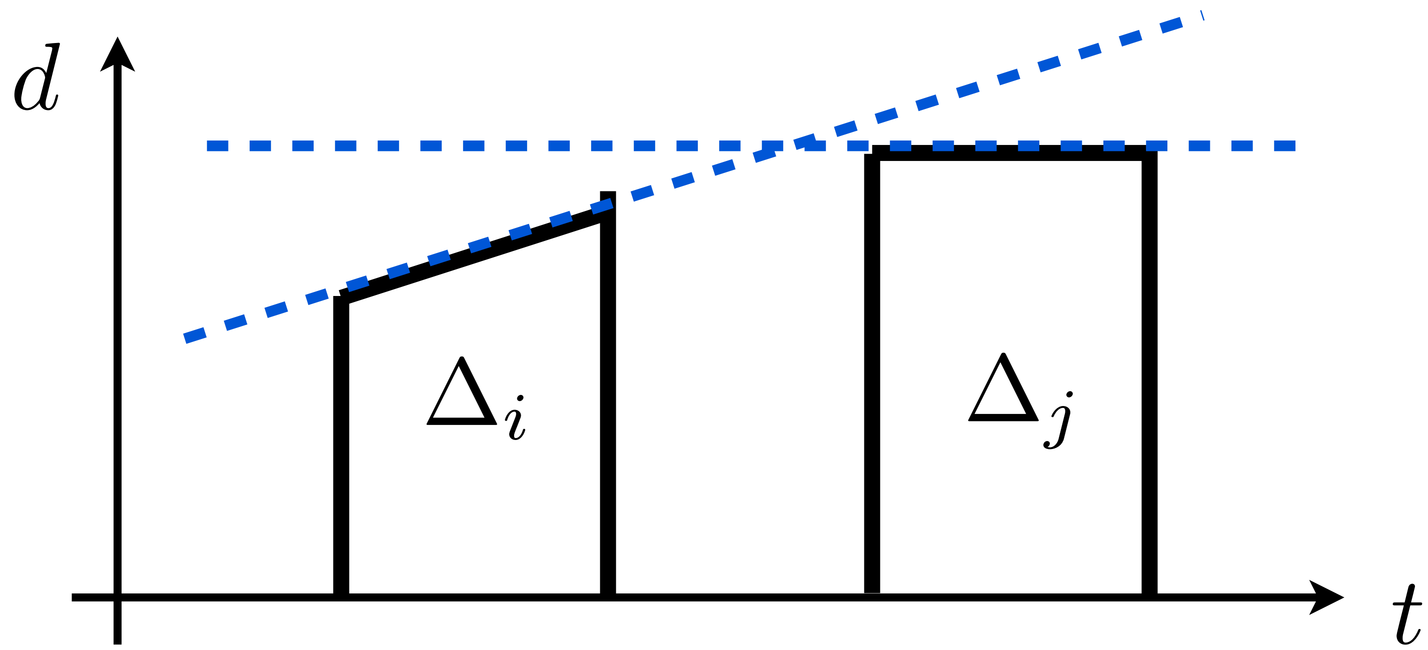
$$h_5(|c_i - c_j|)$$



$$O(\Delta_i, \Delta_j) = h_1 \max(t_j^s - t_i^e, 0) +$$
$$h_2 |\bar{z}_i - z_j| +$$
$$h_3 |k_i - k_j| +$$
$$h_4 (t_i^e - t_i^s + t_j^e - t_j^s) +$$
$$h_5 (|c_i - c_j|)$$

temporal proximity

depths



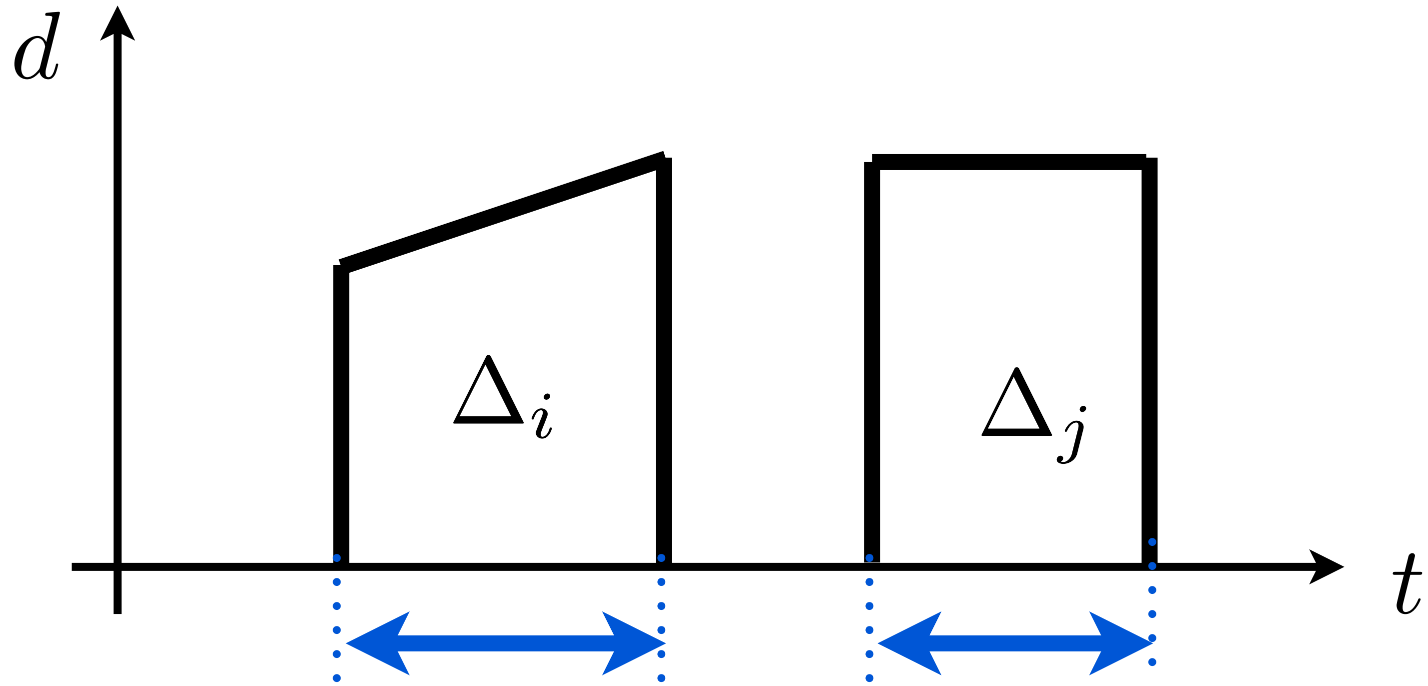
$$O(\Delta_i, \Delta_j) = h_1 \max(t_j^s - t_i^e, 0) +$$ temporal proximity

$$h_2 |\bar{z}_i - z_j| +$$ depths

$$h_3 |k_i - k_j| +$$ slope

$$h_4 (t_i^e - t_i^s + t_j^e - t_j^s) +$$

$$h_5 (|c_i - c_j|)$$



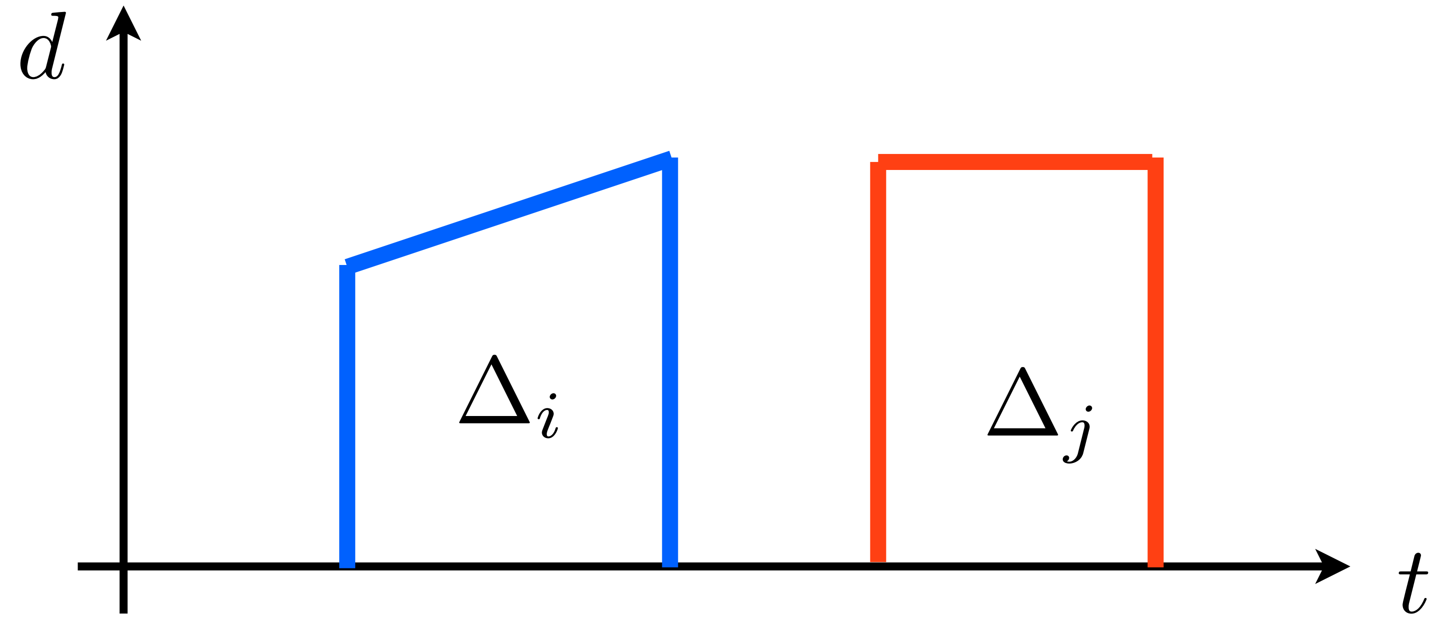
$$O(\Delta_i, \Delta_j) = h_1 \max(t_j^s - t_i^e, 0) +$$ temporal proximity

$$h_2 |\bar{z}_i - z_j| +$$ depths

$$h_3 |k_i - k_j| +$$ slope

$$h_4 (t_i^e - t_i^s + t_j^e - t_j^s) +$$ temporal coverage

$$h_5 (|c_i - c_j|)$$



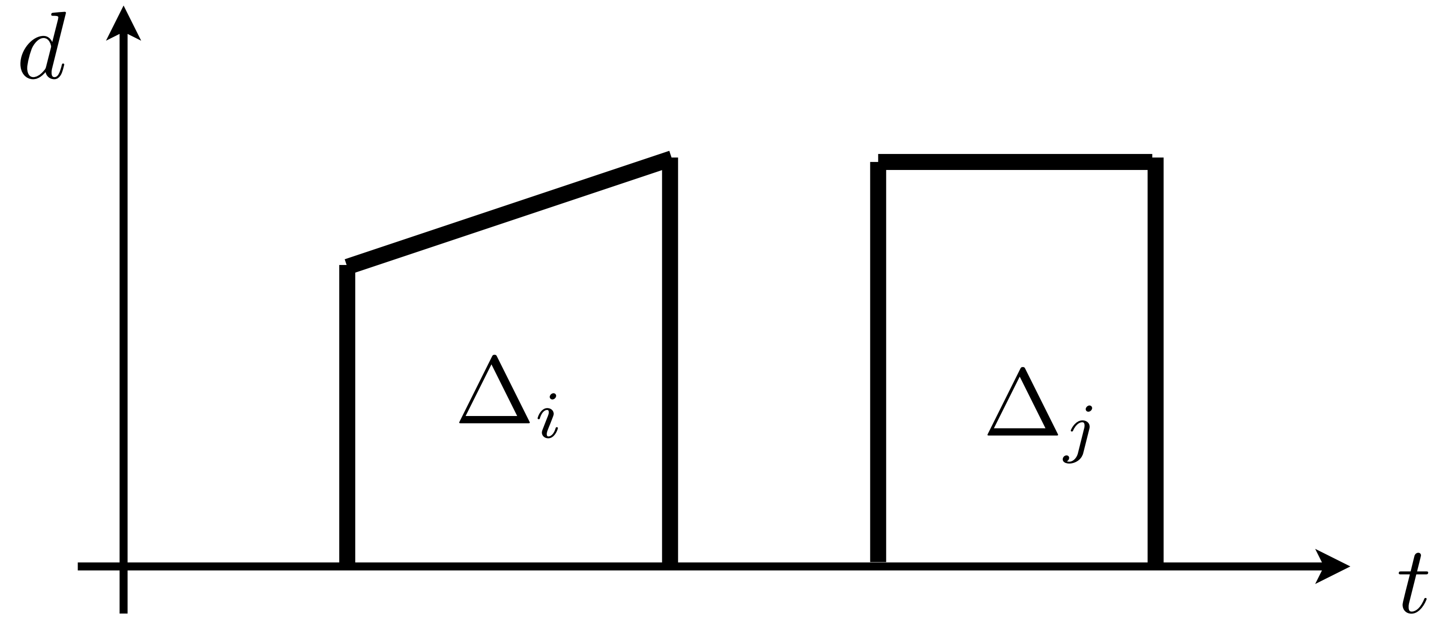
$$O(\Delta_i, \Delta_j) = h_1 \max(t_j^s - t_i^e, 0) +$$ temporal proximity

$$h_2 |\bar{z}_i - z_j| +$$ depths

$$h_3 |k_i - k_j| +$$ slope

$$h_4 (t_i^e - t_i^s + t_j^e - t_j^s) +$$ temporal coverage

$$h_5 (|c_i - c_j|)$$ color norm

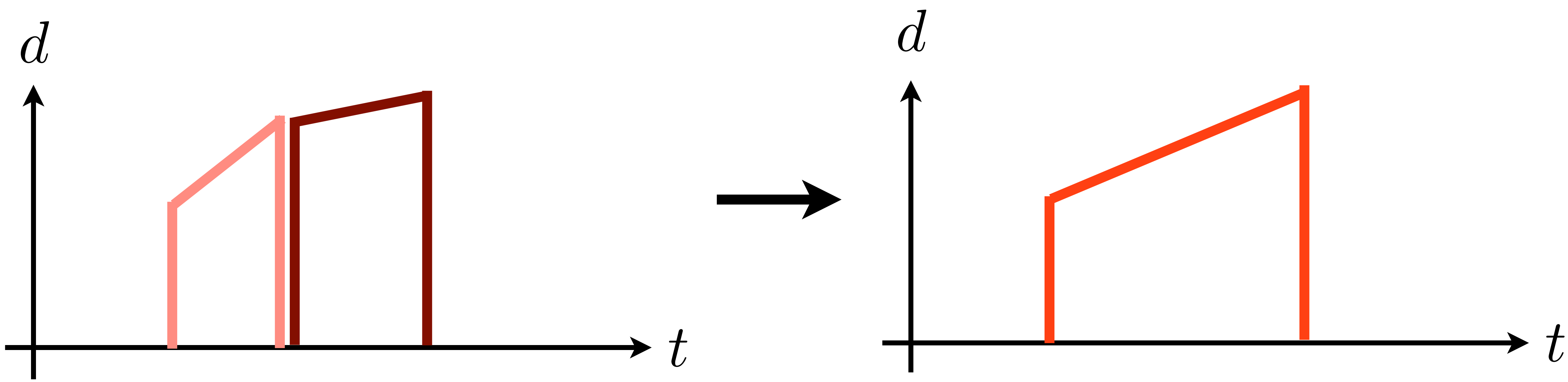


$O(\Delta_i, \Delta_j) = h_1 \max(t_j^s - t_i^e, 0) +$	temporal proximity
$h_2 \lvert \bar{z}_i - z_j \rvert +$	depths
$h_3 \lvert k_i - k_j \rvert +$	slope
$h_4 (t_i^e - t_i^s + t_j^e - t_j^s) +$	temporal coverage
$h_5 (\lvert c_i - c_j \rvert)$	color norm

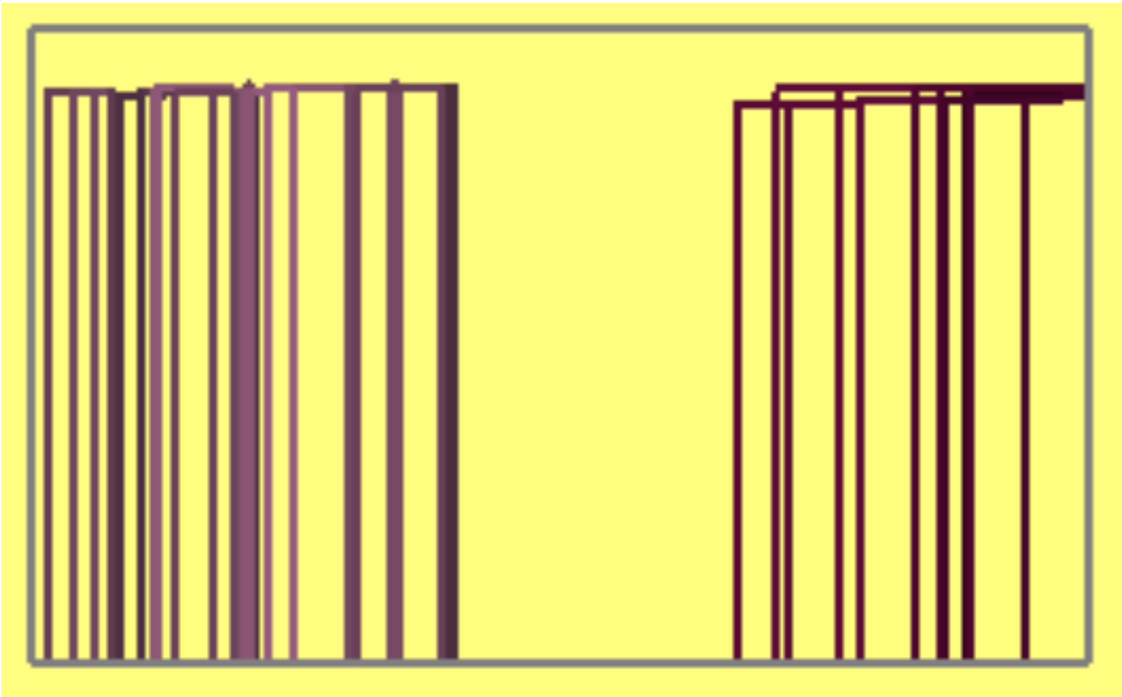
weights

Interval merging

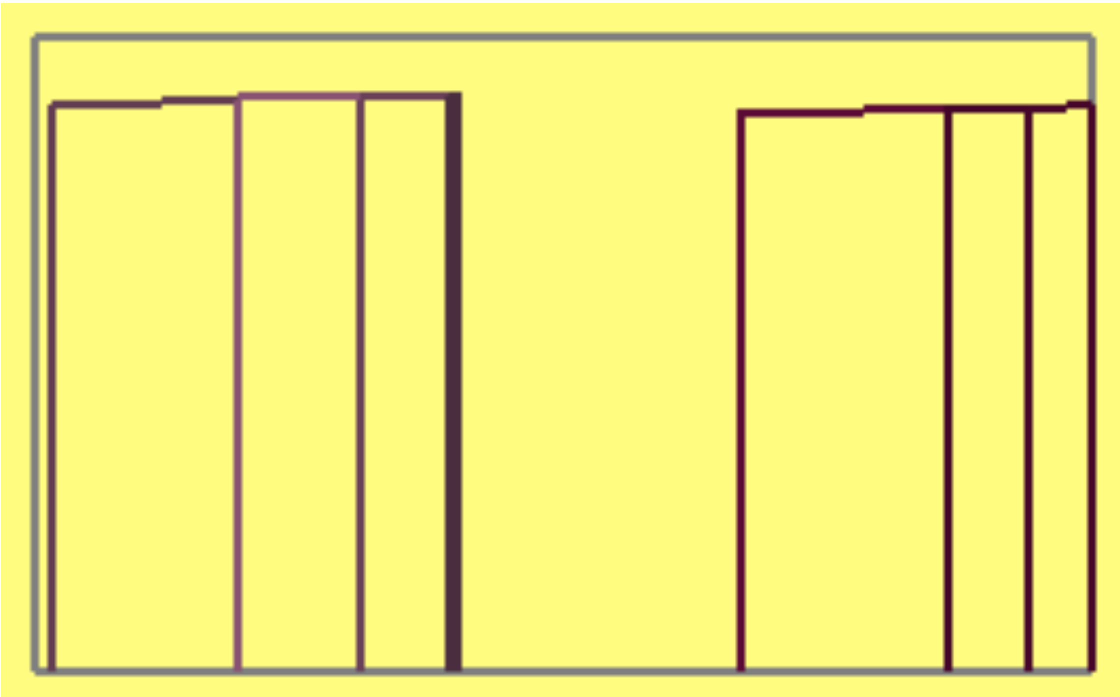
- Merge interval pair with lowest Oracle value
- Combine temporal coverage
- Blend attributes



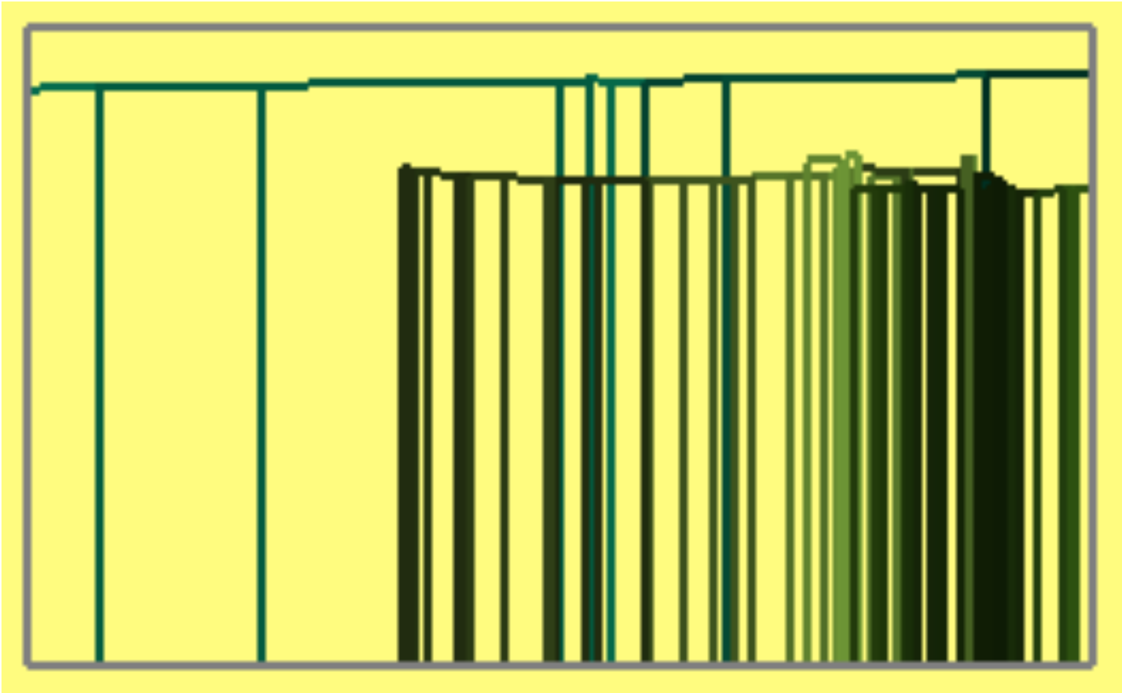
Compression with occlusion



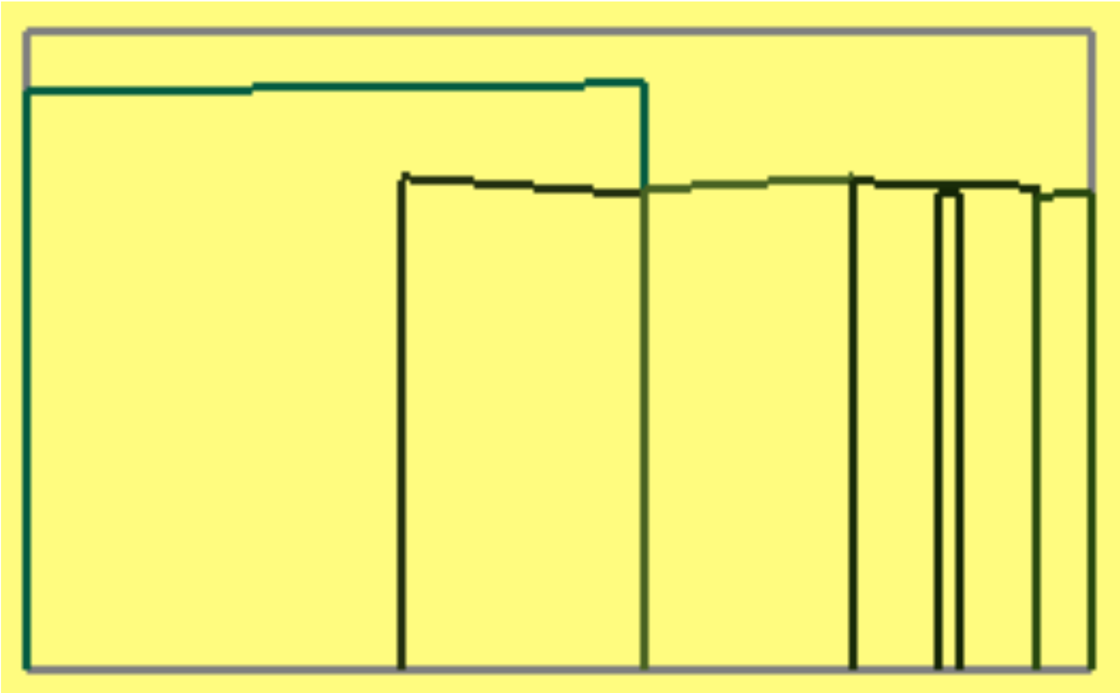
39 intervals



8 intervals

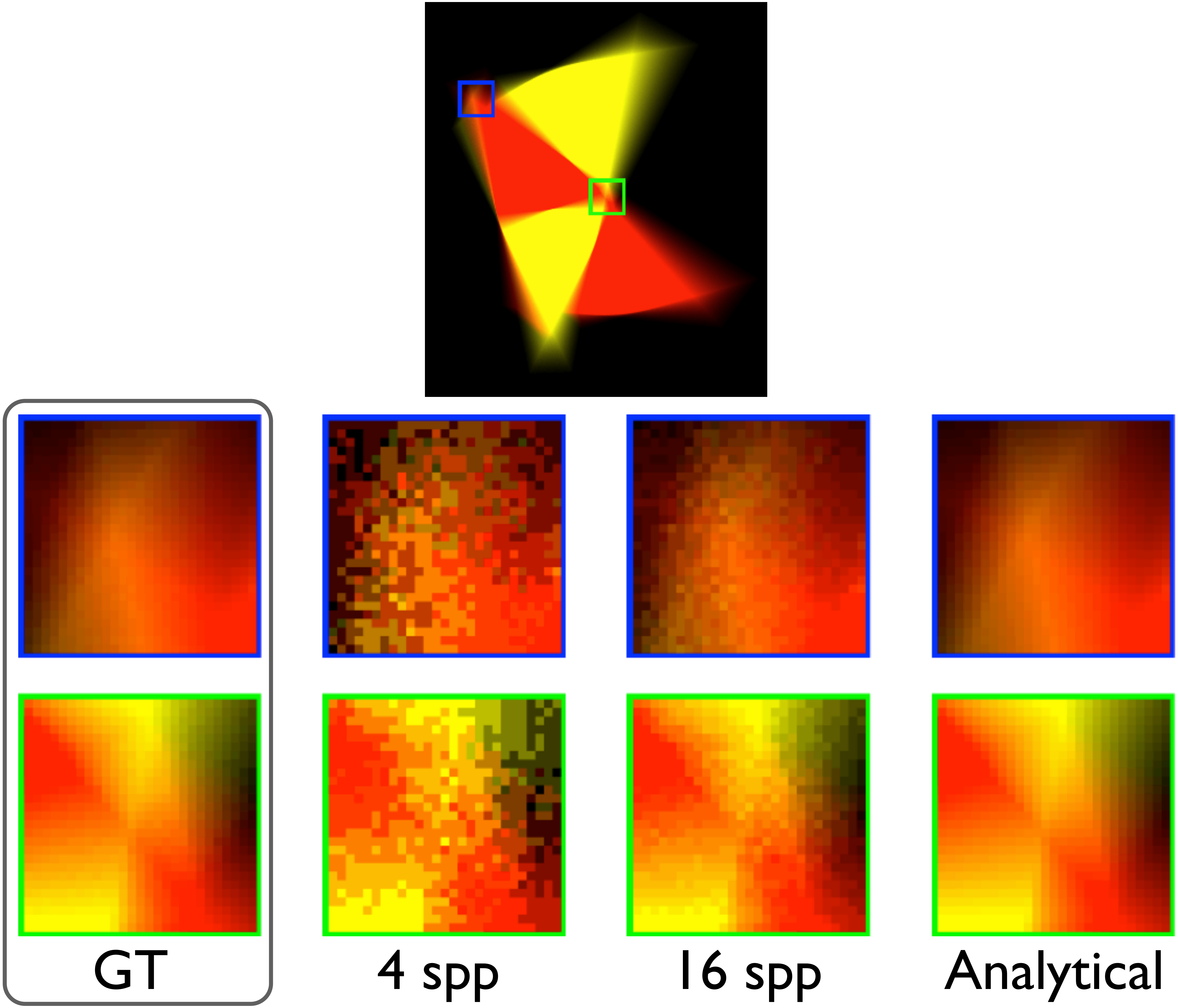


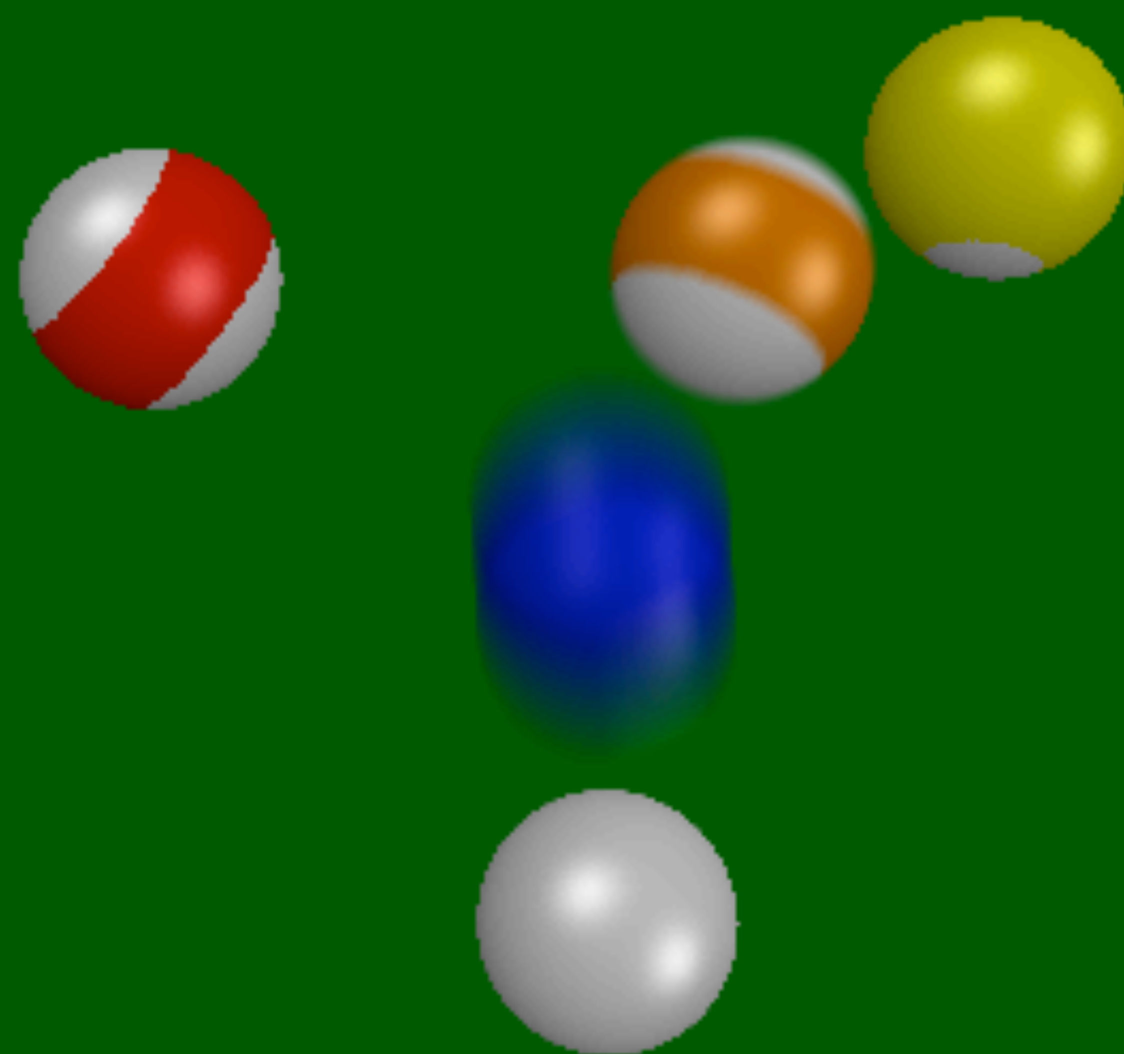
77 intervals



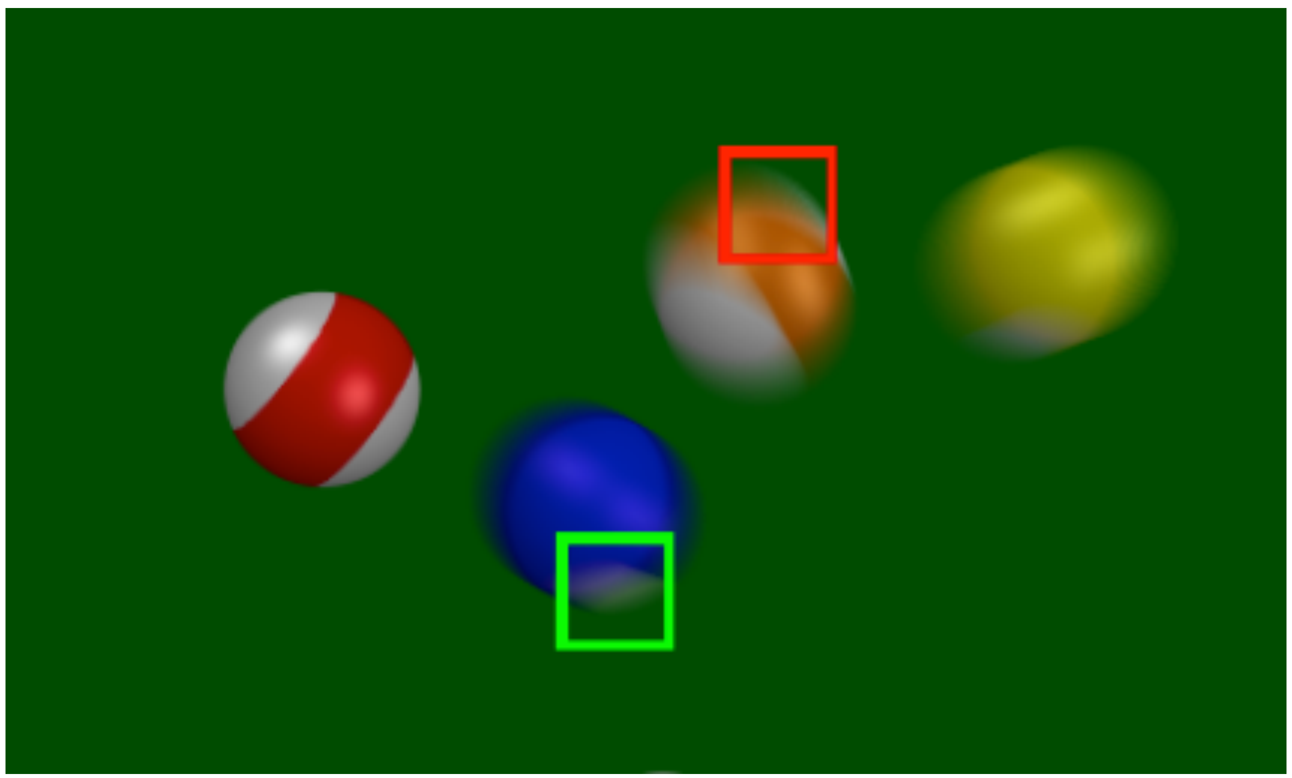
8 intervals

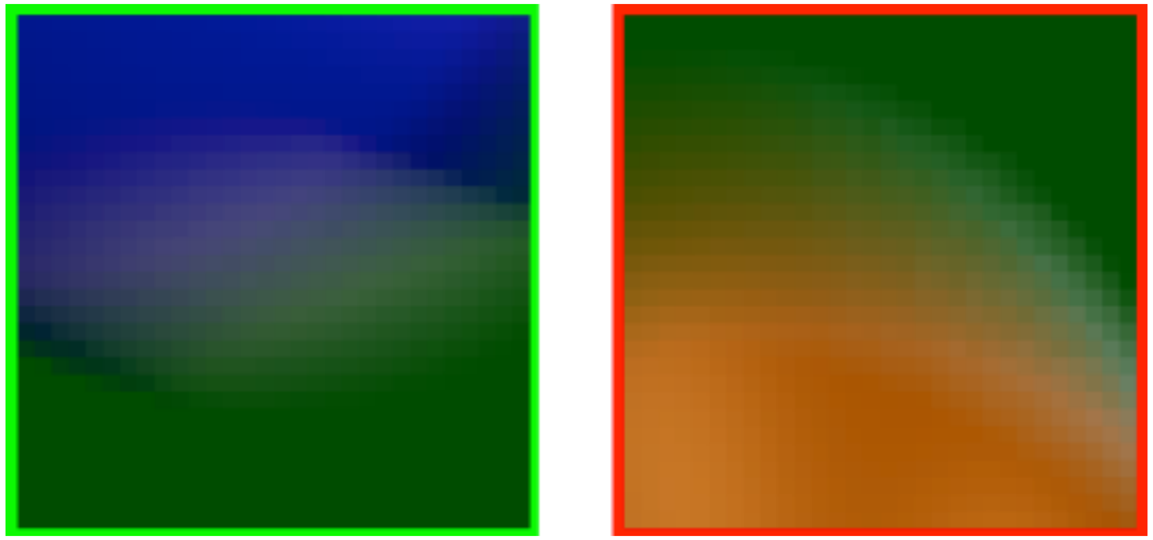
Results


GT
4 spp
16 spp
Analytical

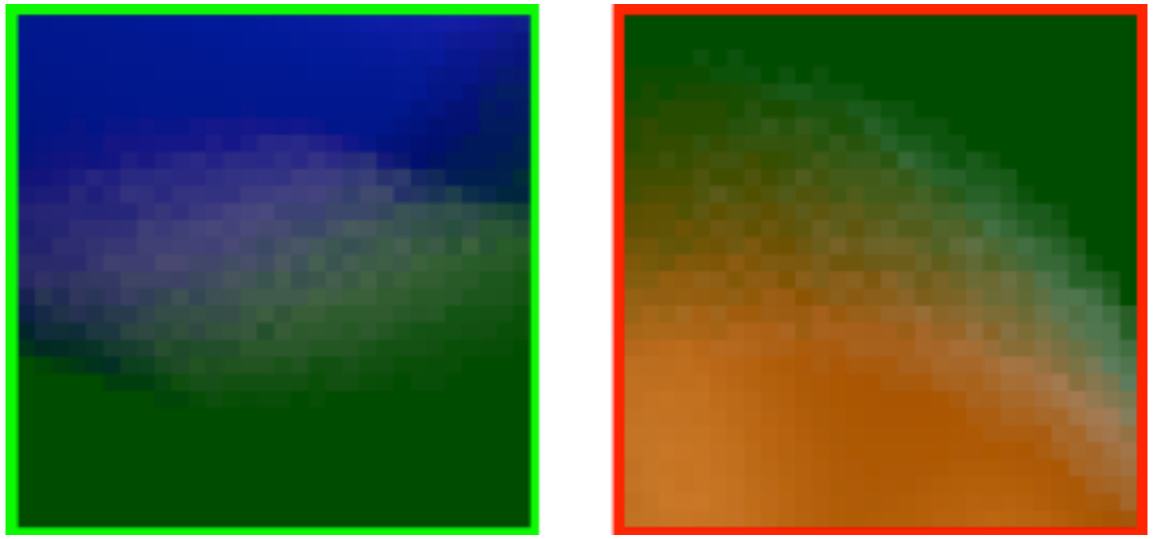


Compression: 8

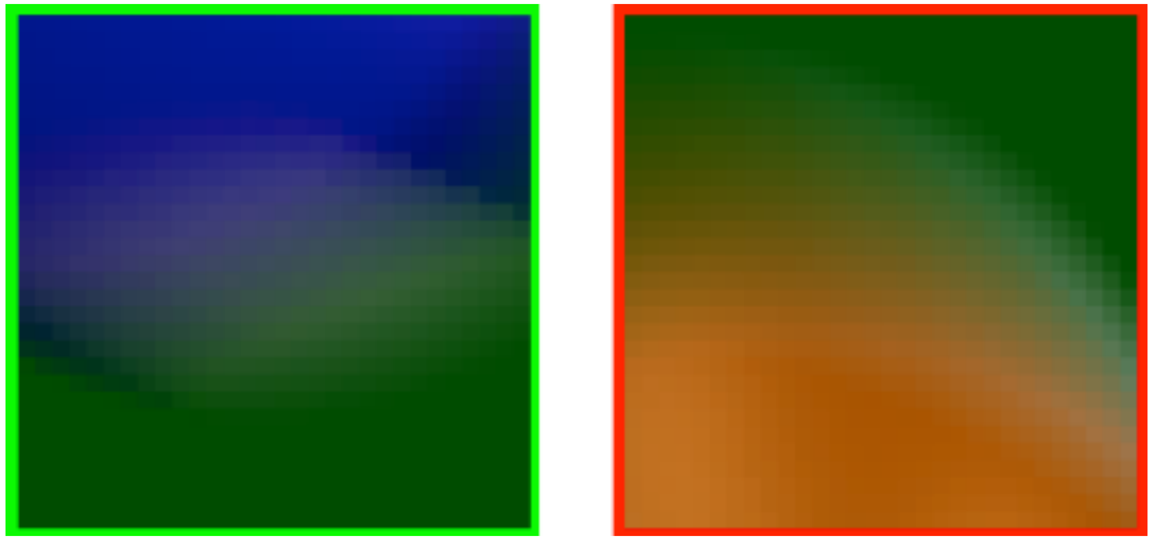




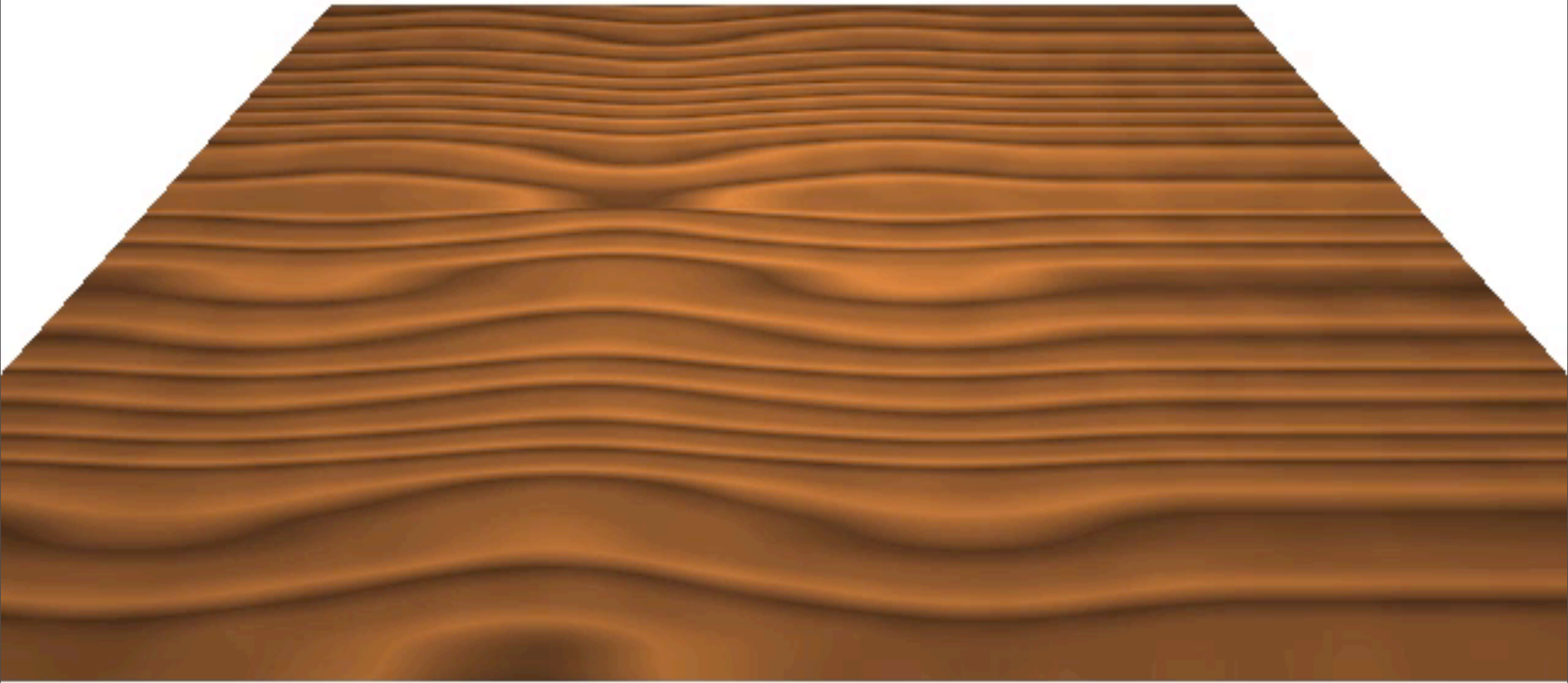
Ground Truth



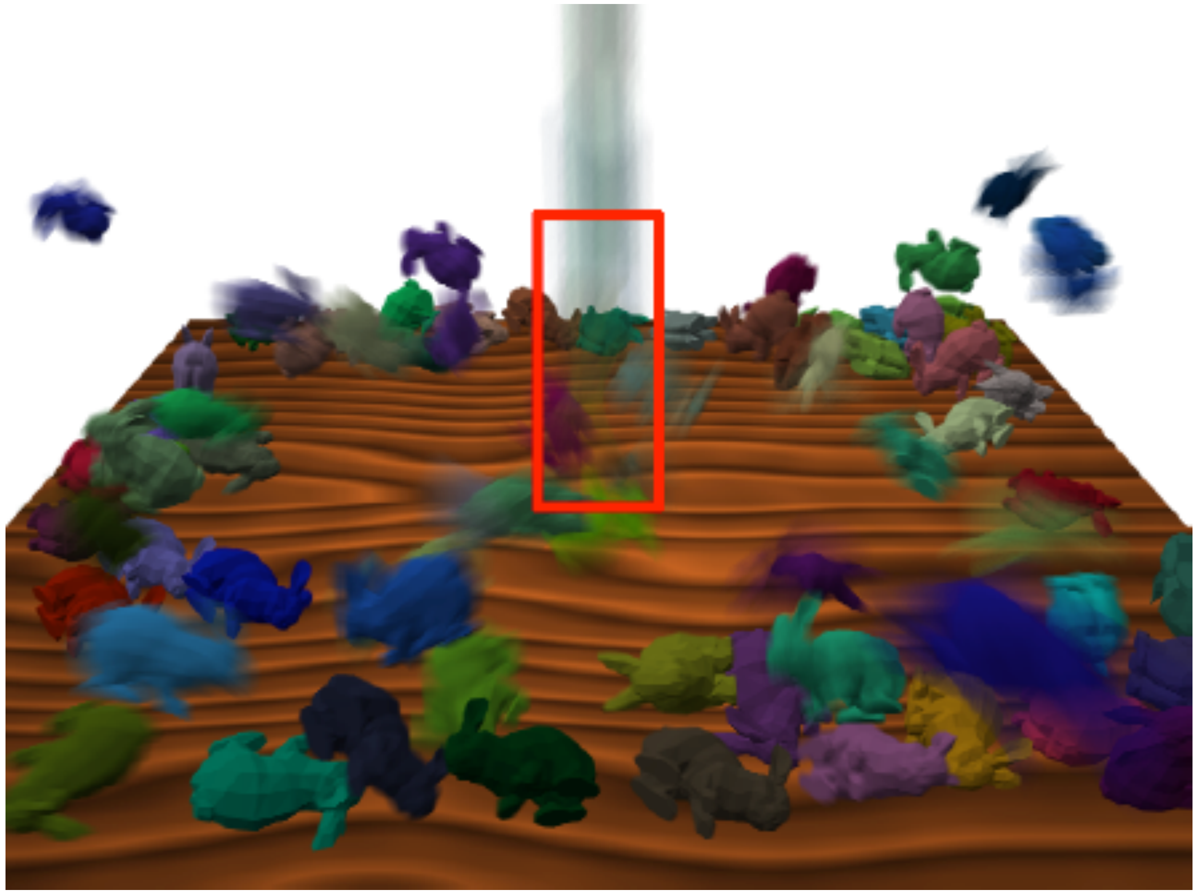
Stochastic 12 spp

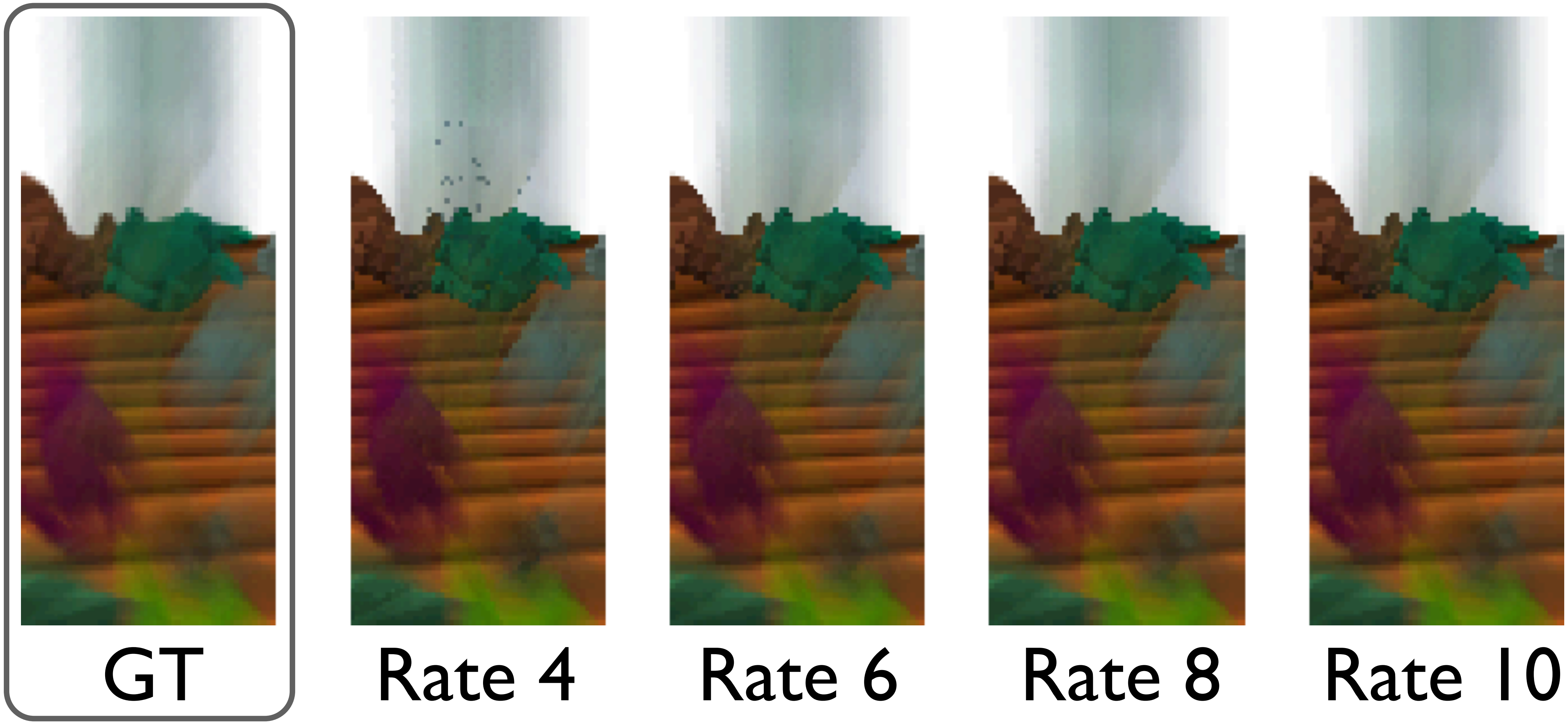


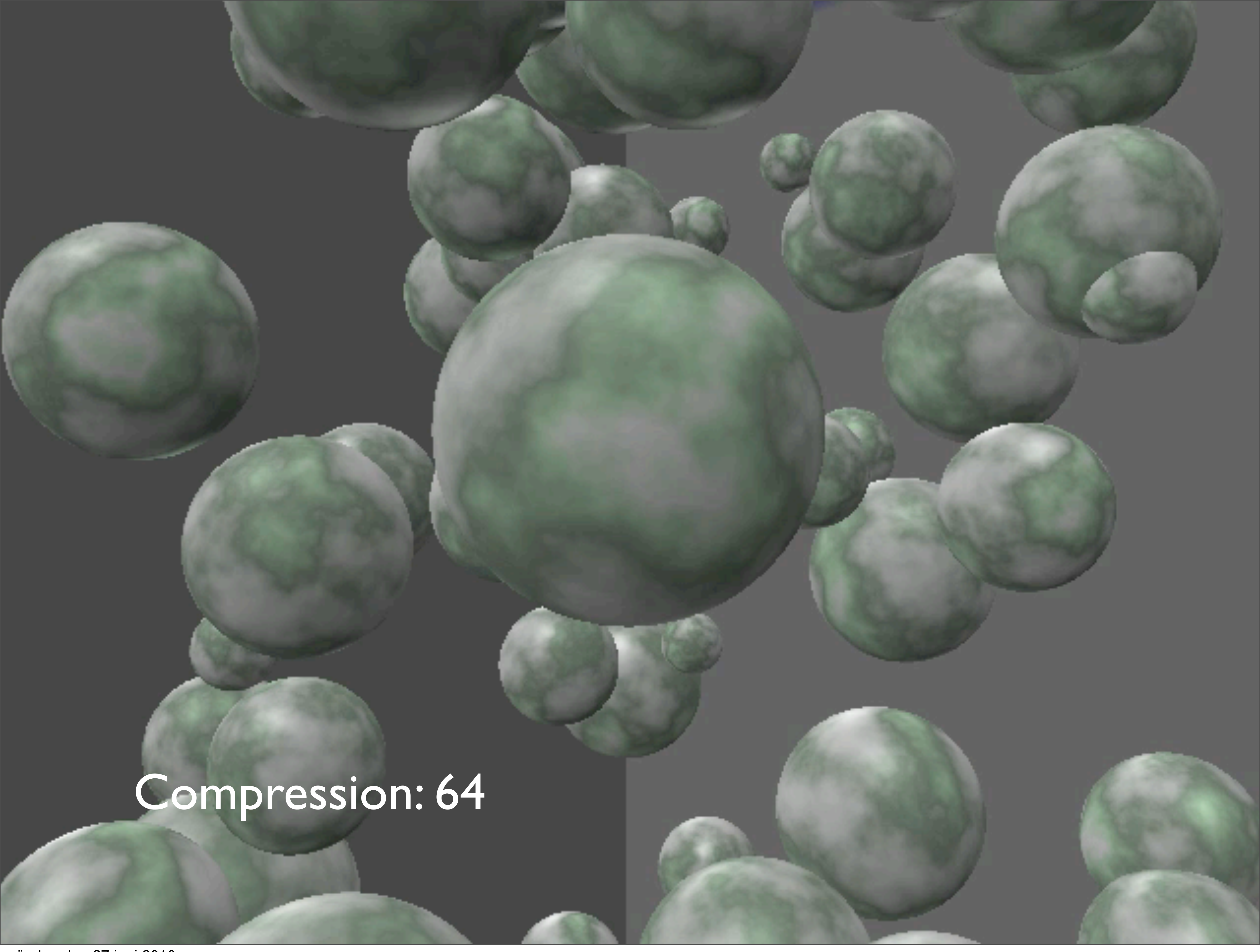
Analytical, compression 8



Compression Rates






Compression: 64

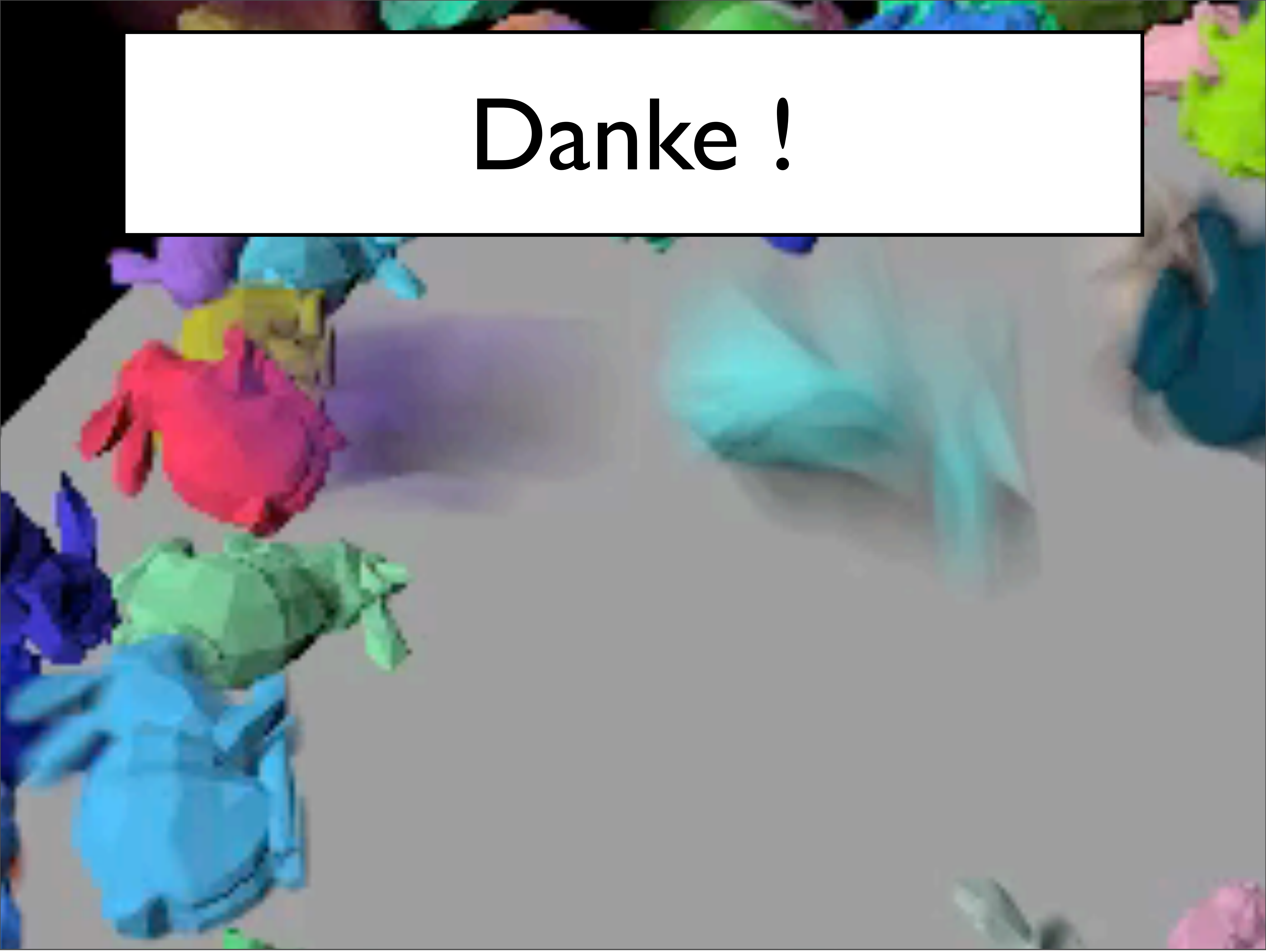
Algorithm summary

- Time-dependent edge equations to compute exact exposure intervals
- Visibility management by using a linear approximation of a cubic rational depth function
- Blur translucent geometry by performing alpha-blending extended in time
- We propose a compression algorithm to address fixed memory requirements

Future work

- GPU implementation (in progress)
- Motion blurred shadows (in progress)
- Apply decoupled shading
- Analytical spatial anti-aliasing
- Higher order motion

Danke !

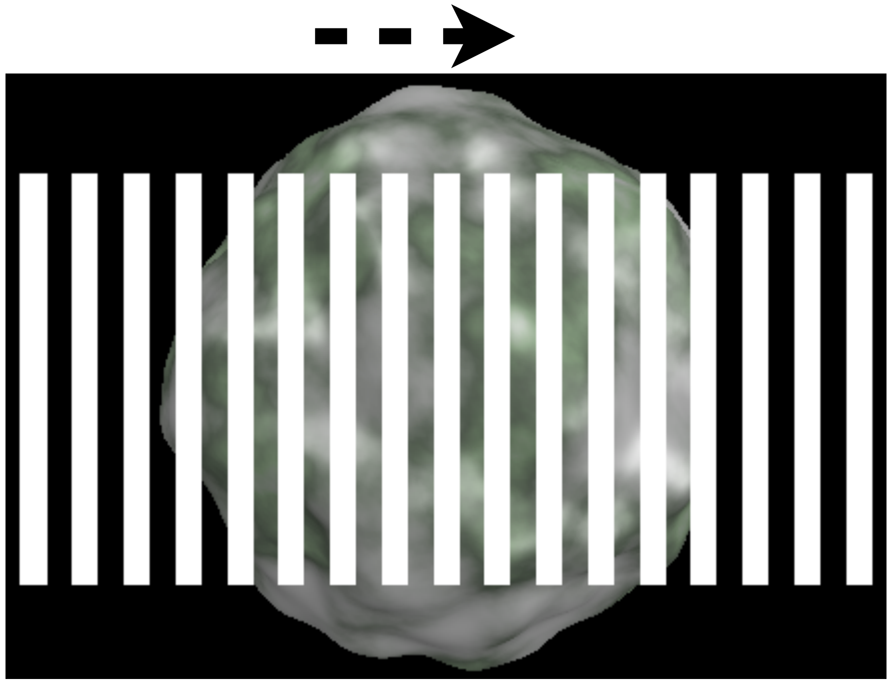


Danke !

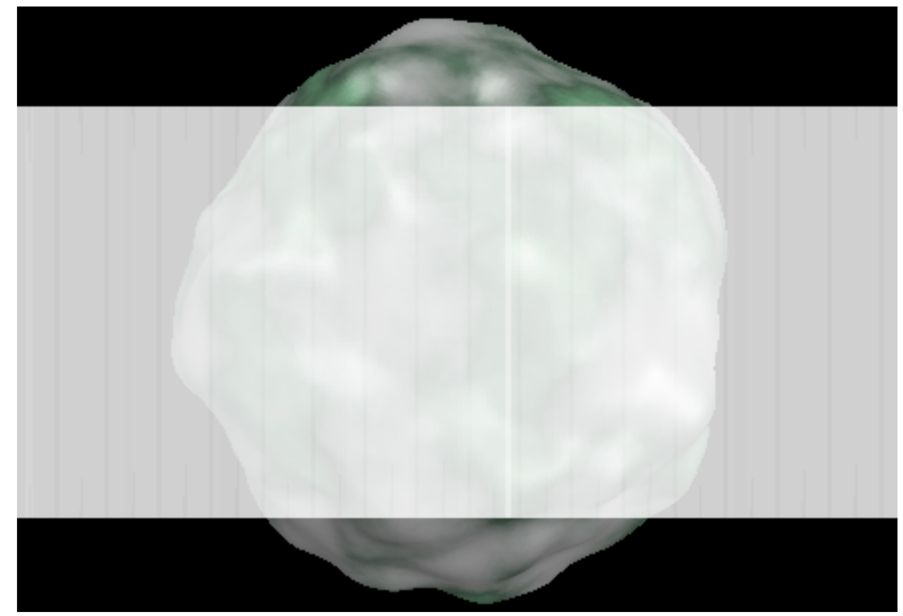
* No bunnies were harmed during the research

Compression failure case

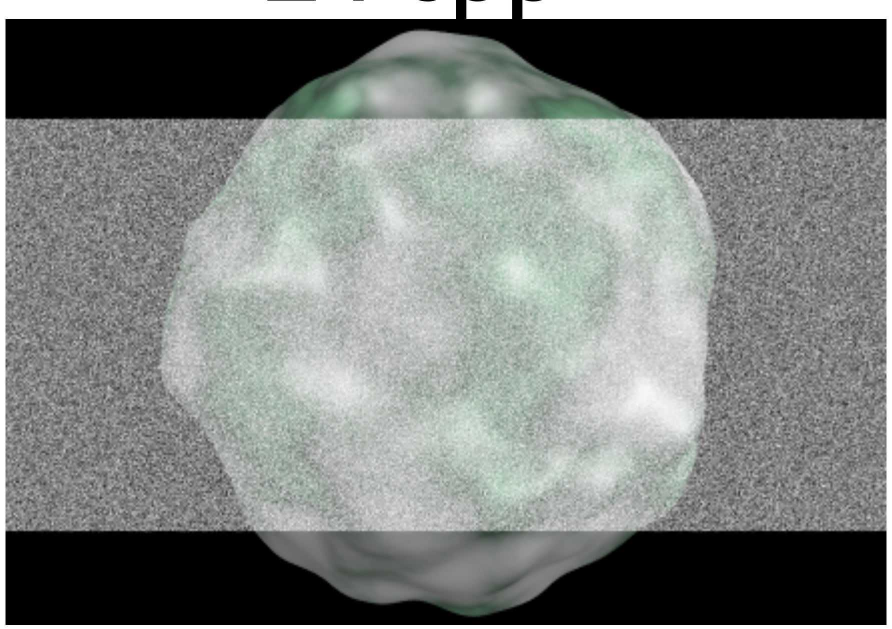
Fast moving fence



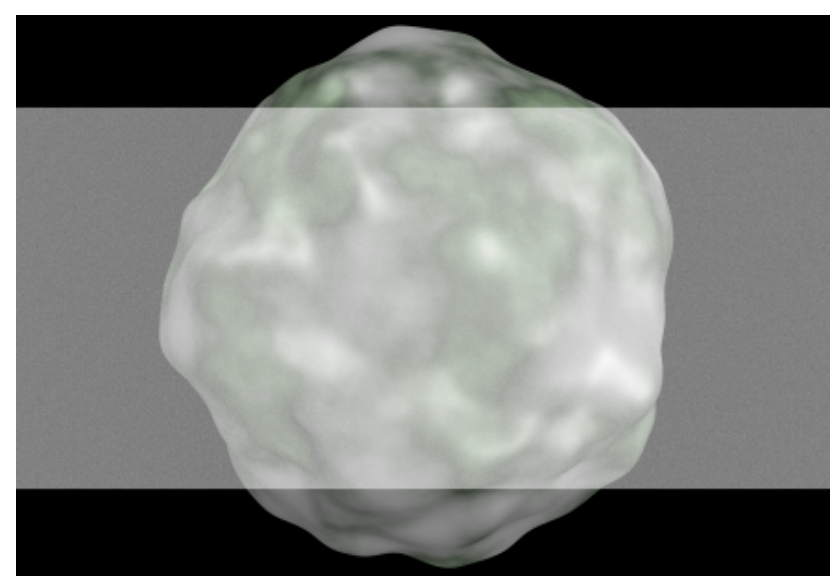
Analytical 16 intervals



24 spp



256 spp



Depth approximation error