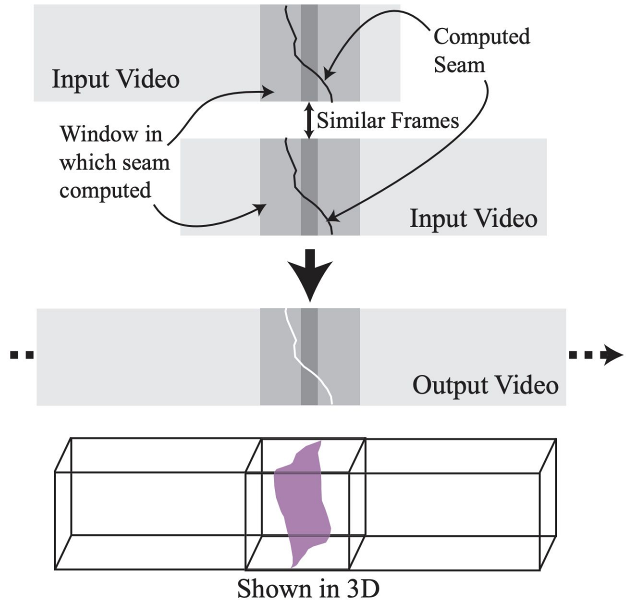
Automated Video Looping with Progressive Dynamism

CS448V: Lecture 5

Background

Schodl et al.



Kwatra et al.

Pixel-Based Looping

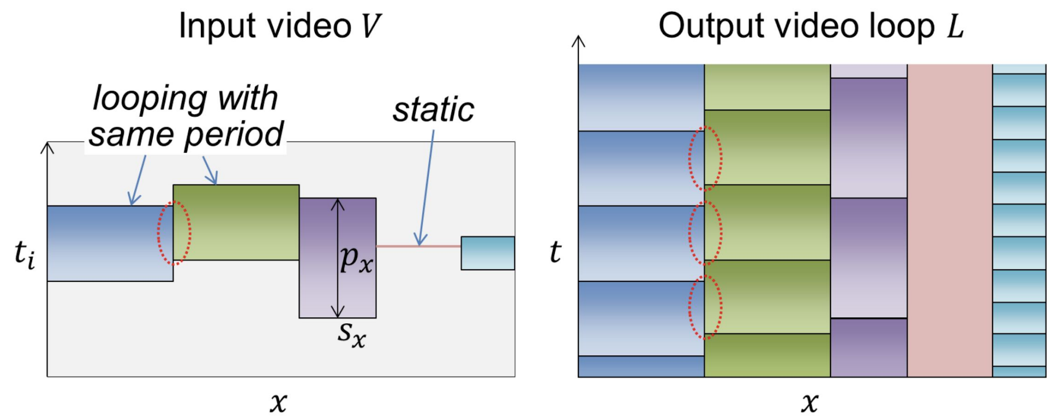
Pixel-Based Looping

Overview

1. Per-Pixel Loops
2. Finding a Video Loop
3. Progressive Video Loops

Per-Pixel Loops

Problem Statement



$$L(x,t) = V\left(x, \phi(x,t)\right), \quad t \geq 0,$$

Problem Statement

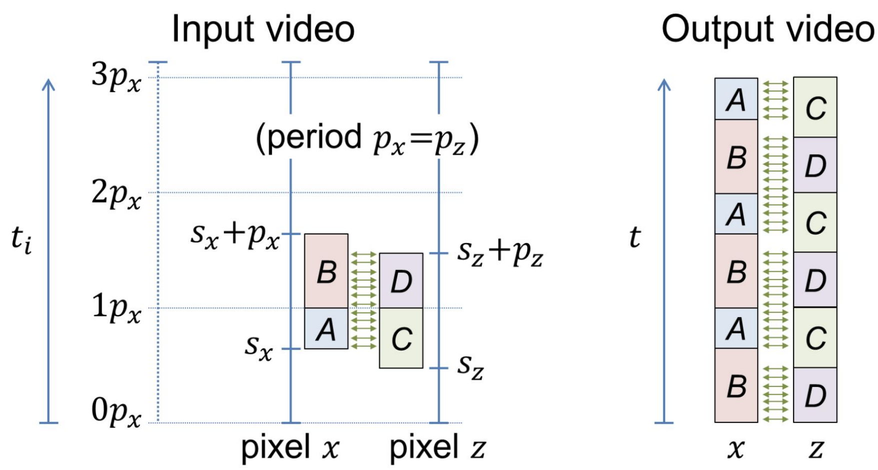
$\phi(x,t)$ is defined by s_x, p_x

Problem Statement

$$\phi(x, t) = s_x + (t - s_x) \bmod p_x$$

When t = 0,

$\phi(x, t) = s_x + (-s_x) \bmod p_x$
$= s_x + (-s_x) + p_x$
$= p_x$



Problem Statement

$$L(x, t) = V\left(x, \phi(x, t)\right), \quad t \geq 0,$$

$\phi(x, t)$ is defined by s_x, p_x

Loops for the entire video can be defined by:

$$\mathbf{s} = \{s_x\} \qquad \mathbf{p} = \{p_x\}$$

"Energy": Cost of a Solution

A solution consists of: $\mathbf{s} = \{s_x\} \quad \mathbf{p} = \{p_x\}$

Want to minimize:

$$E(\mathbf{s}, \mathbf{p}) = E_{\text{consistency}}(\mathbf{s}, \mathbf{p}) + E_{\text{static}}(\mathbf{s}, \mathbf{p})$$

Spatiotemporal consistency

Penalty for choosing static loops

Spatiotemporal Consistency

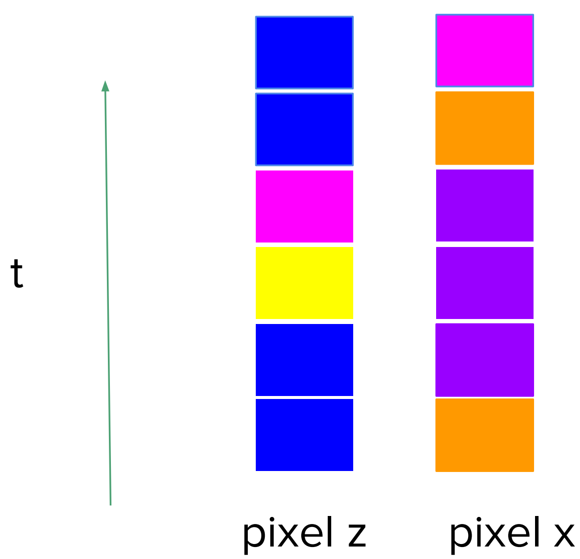
$$E_{\text{consistency}}(\mathbf{s}, \mathbf{p}) = \beta E_{\text{spatial}}(\mathbf{s}, \mathbf{p}) + E_{\text{temporal}}(\mathbf{s}, \mathbf{p})$$

Spatial Consistency

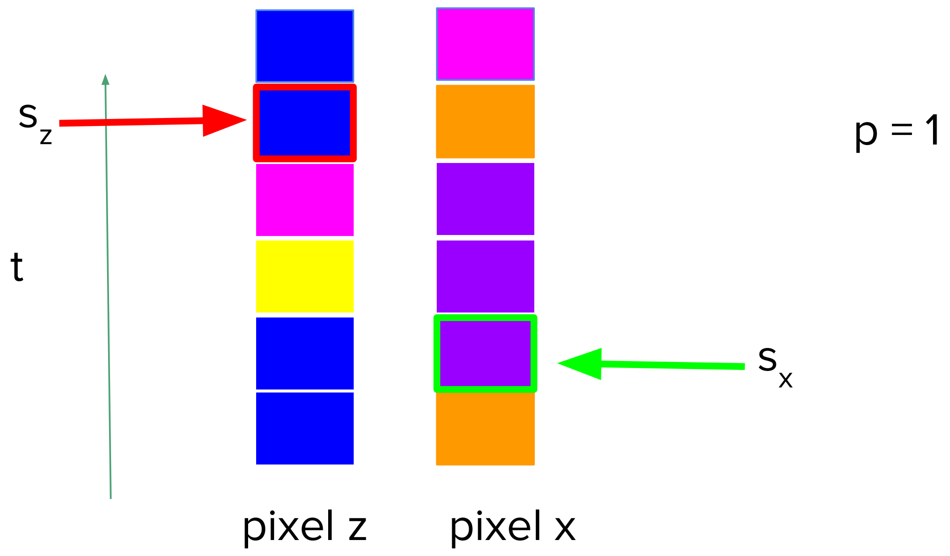
$$E_{\text{spatial}}(\mathbf{s}, \mathbf{p}) = \sum_{\|x-z\|=1} \Psi(x, z)$$

Compatibility of
adjacent pixels x, z
over loop

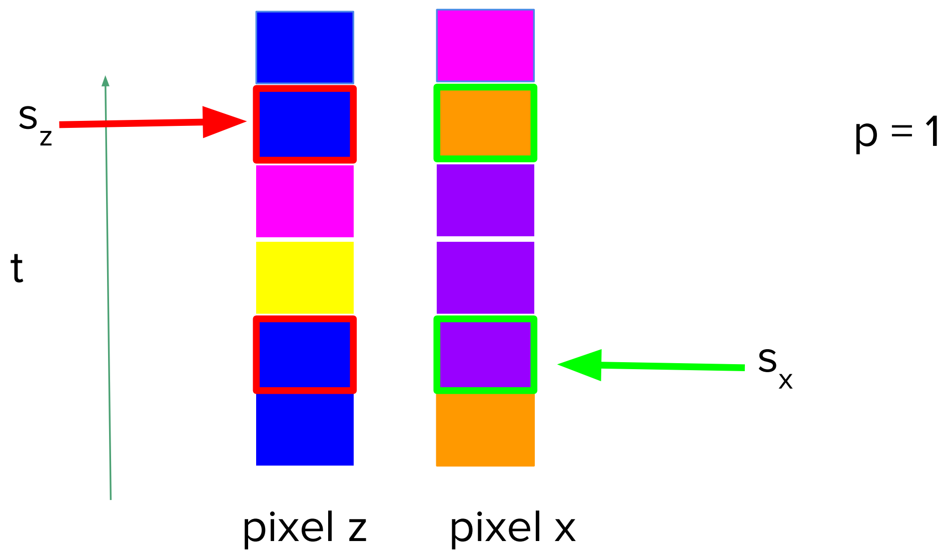
Spatial Consistency



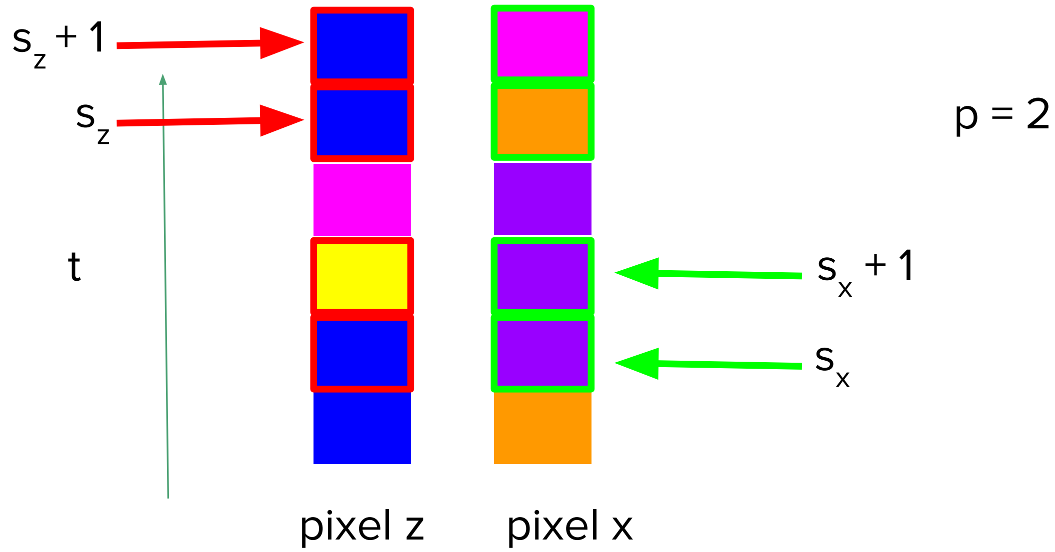
Spatial Consistency



Spatial Consistency



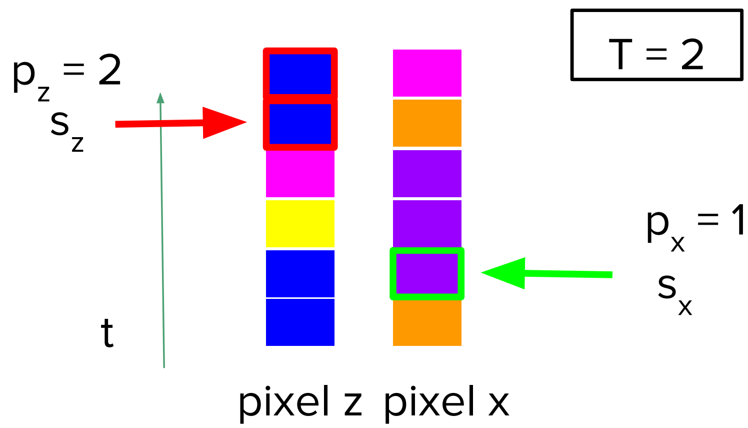
Spatial Consistency



Spatial Consistency

$$\Psi(x,z) = \frac{1}{T}\sum_{t=0}^{T-1}\begin{pmatrix}\|V(x,\phi(x,t)) - V(x,\phi(z,t))\|^2 + \\ \|V(z,\phi(x,t)) - V(z,\phi(z,t))\|^2\end{pmatrix}$$

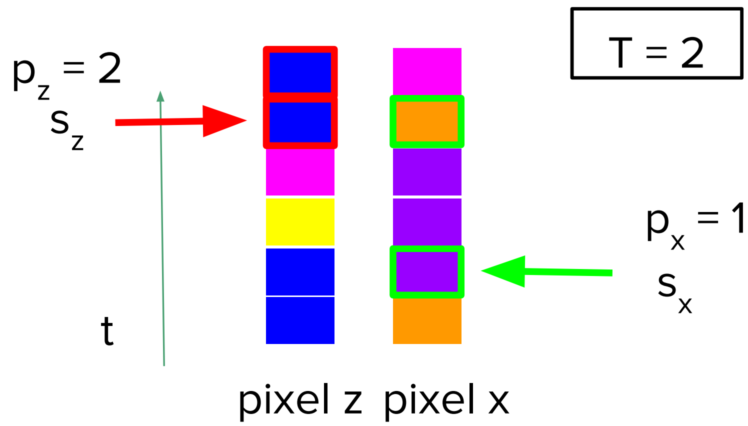
$$T = \mathrm{LCM}(p_x, p_z)$$



Spatial Consistency

$$\Psi(x,z) = \frac{1}{T}\sum_{t=0}^{T-1}\begin{pmatrix}\|V(x,\phi(x,t)) - V(x,\phi(z,t))\|^2 + \\ \|V(z,\phi(x,t)) - V(z,\phi(z,t))\|^2\end{pmatrix}$$

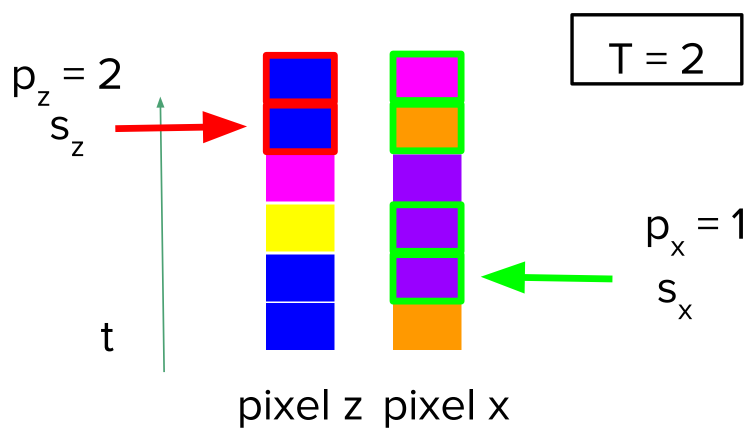
$$T = \mathrm{LCM}(p_x, p_z)$$



Spatial Consistency

$$\Psi(x,z) = \frac{1}{T}\sum_{t=0}^{T-1}\begin{pmatrix}\|V(x,\phi(x,t)) - V(x,\phi(z,t))\|^2 + \\ \|V(z,\phi(x,t)) - V(z,\phi(z,t))\|^2\end{pmatrix}$$

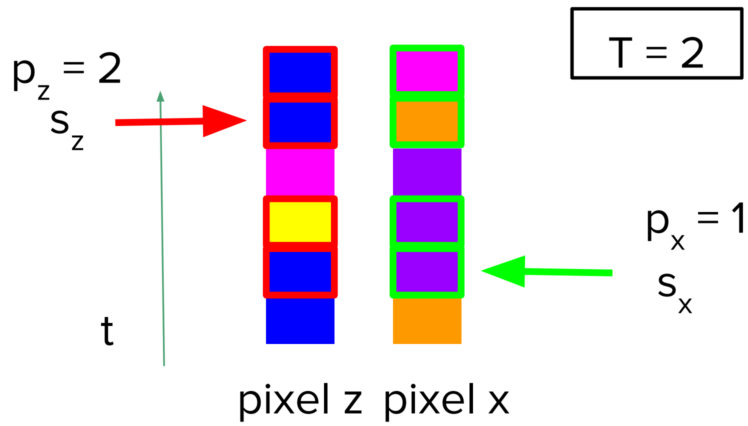
$$T = \mathrm{LCM}(p_x, p_z)$$



Spatial Consistency

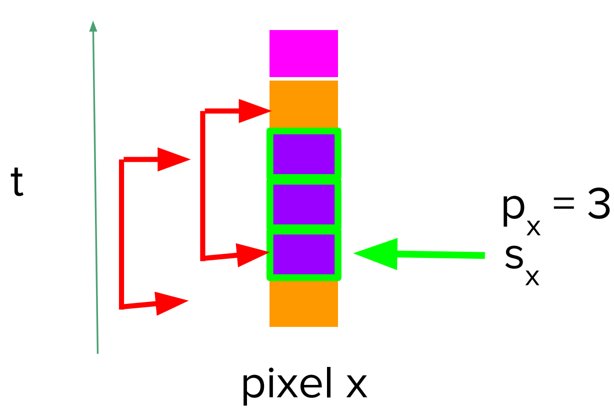
$$\Psi(x,z) = \frac{1}{T}\sum_{t=0}^{T-1}\begin{pmatrix}\|V(x,\phi(x,t)) - V(x,\phi(z,t))\|^2 + \\ \|V(z,\phi(x,t)) - V(z,\phi(z,t))\|^2\end{pmatrix}$$

$$T = \mathrm{LCM}(p_x, p_z)$$



Temporal Consistency

$$E_{\text{temporal}} = \sum_x \begin{pmatrix}\|V(x,s_x) - V(x,s_x+p_x)\|^2 + \\ \|V(x,s_x-1) - V(x,s_x+p_x-1)\|^2\end{pmatrix}$$



$$E(\mathbf{s}, \mathbf{p}) = E_{\text{consistency}}(\mathbf{s}, \mathbf{p}) + \boxed{E_{\text{static}}(\mathbf{s}, \mathbf{p})}$$

Spatiotemporal consistency

Penalty for choosing static loops

Static Loop Penalty

$$E_{\text{static}} = \sum_{\boxed{x | p_x = 1}} E_{\text{static}}(x)$$

$$E_{\text{static}}(x) = \boxed{c_{\text{static}}} \boxed{\gamma_{\text{static}}}$$

Constant penalty for assigning a pixel as static

Scale Factor

Attenuating Static Cost

If original pixel had HIGH variance, a static loop is LESS natural

If original pixel had LOW variance, a static loop is more acceptable

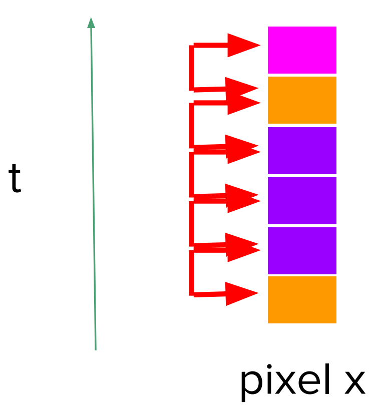
Attenuating Static Penalty

γ_{static} ↔ Measure of temporal variation

$$E_{\text{static}}(x) = c_{\text{static}} \min\left(1, \lambda_{\text{static}} \operatorname*{MAD}_{t_i} \left\| N(x, t_i) - N(x, t_i + 1) \right\|\right)$$

Median Absolute Deviation

Gaussian weighted neighborhood



Attenuating **Spatiotemporal** Consistency Cost

If original pixel had LOW variance, a loop with high variance is MORE perceptible

If original pixel had HIGH variance, a loop with high variance is LESS perceptible

Attenuating Spatiotemporal Consistency Cost

$$E_{\text{spatial}}(\mathbf{s},\mathbf{p}) = \sum_{\|x-z\|=1} \Psi(x,z)\boxed{\gamma_s(x,z)}$$

If spatial variance is high, γ_s should be small.

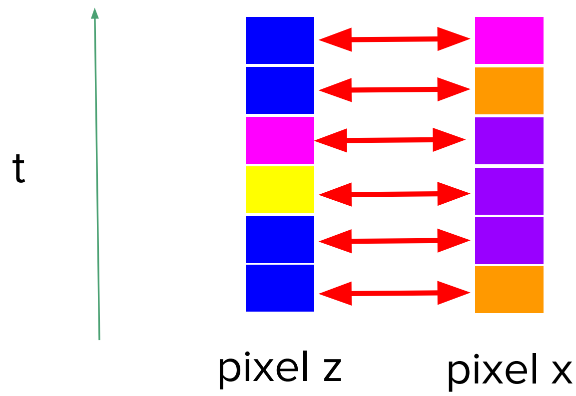
$$E_{\text{temporal}} = \sum_{x} \begin{pmatrix} \|V(x,s_x) - V(x,s_x+p_x)\|^2 + \\ \|V(x,s_x-1) - V(x,s_x+p_x-1)\|^2 \end{pmatrix} \boxed{\gamma_t(x)}$$

If temporal variance is high, γ_t should be small.

Attenuating Spatial Consistency Cost

$$\gamma_s(x,z) = 1/\left(1 + \lambda_s \operatorname*{MAD}_{t_i} \left\|V(x,t_i) - V(z,t_i)\right\|\right)$$

Measure of spatial variance

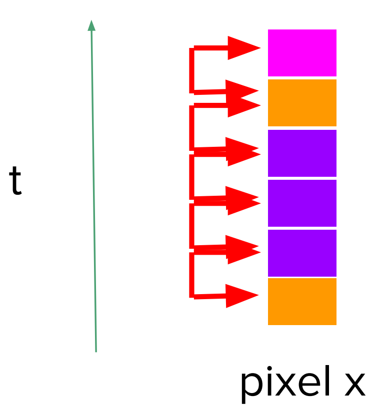


If spatial variance is high, γ_s should be small.

Attenuating Temporal Consistency Cost

$$\gamma_t(x) = 1/\left(1 + \lambda_t \operatorname*{MAD}_{t_i} \left\|V(x,t_i) - V(x,t_i+1)\right\|\right)$$

Measure of temporal variance



If temporal variance is high, γ_t should be small.

Optimization: Solving for $\mathbf{s} = \{ s_x \}$ and $\mathbf{p} = \{ p_x \}$

High-Level Goal

Each node is a pixel

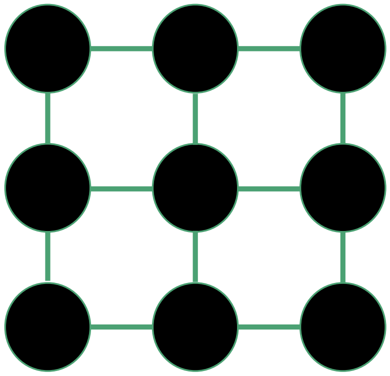
Want to assign each pixel a value (s_x, p_x)

where $s_x \in \mathbf{s}$, $p_x \in \mathbf{p}$

Such that the "total energy" is minimized:

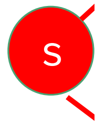
$$E(\mathbf{s}, \mathbf{p}) = E_{\text{consistency}}(\mathbf{s}, \mathbf{p}) + E_{\text{static}}(\mathbf{s}, \mathbf{p})$$

Can formulate as Multilabel graph cut problem



Review: Binary Graph Cut

- Each node: a pixel
- Goal: Partition nodes into two groups
- Want to minimize:
  - Energy = cost of partition
  - Could be formulated as max flow, min cut problem
- Global minima found in polynomial time



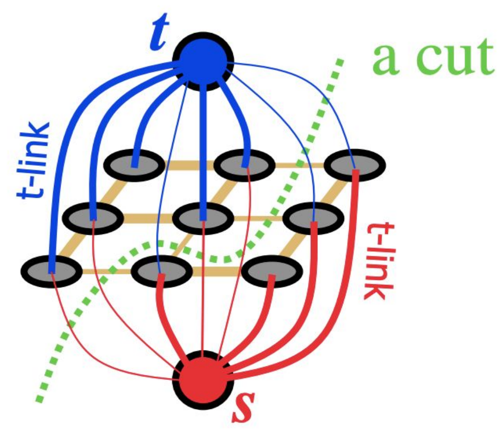


Image credit: http://www.csd.uwo.ca/~yuri/Presentations/ECCV06_tutorial_partI_yuri.pdf

Review: Binary Graph Cut

- Each node: a pixel
- Goal: Partition nodes into two groups
- Want to minimize:
  - Energy = cost of partition
  - Could be formulated as max flow, min cut problem
- Global minima found in polynomial time

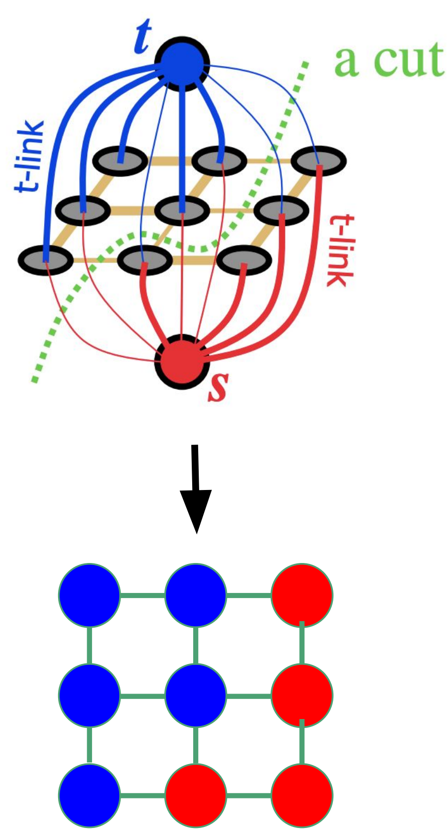
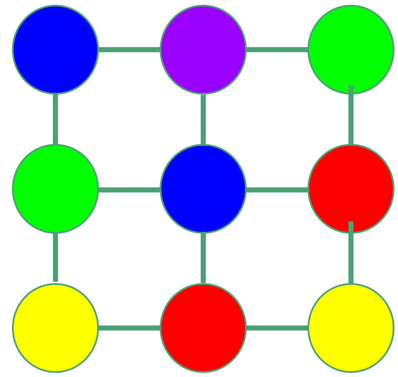


Image credit: http://www.csd.uwo.ca/~yuri/Presentations/ECCV06_tutorial_partI_yuri.pdf

Multilabel Graph Cut

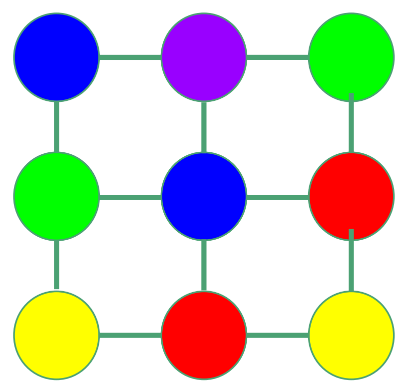
- Generalization of Binary Graph Cut (2 labels)
- NP-hard problem (3 or more labels)
- Alpha-expansion approximation algorithm
    - Within factor of 2 of global minima

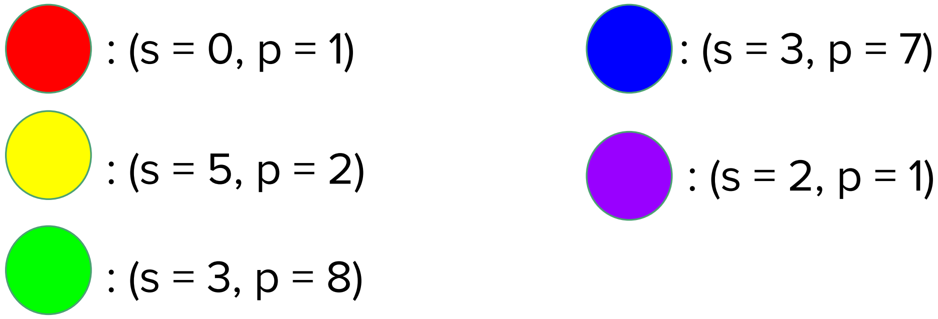


Example: 5 labels

Multilabel Graph Cut

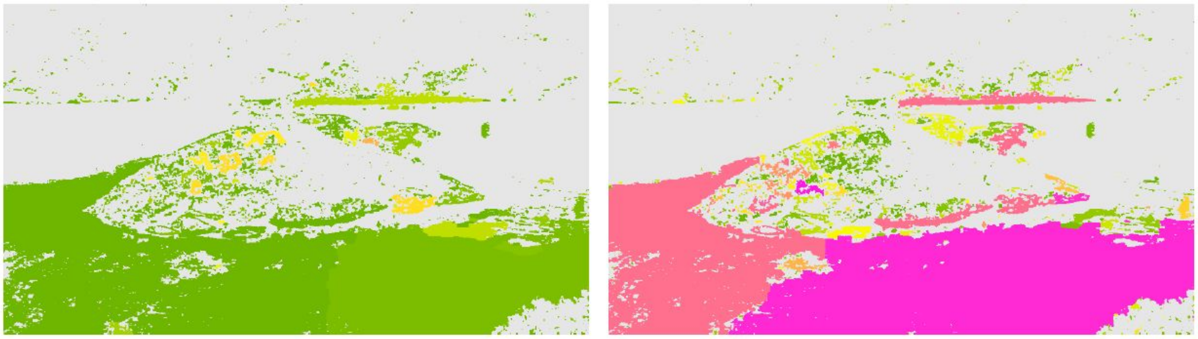
- Assign each pixel a label: (s_x, p_x)
- From a set of candidate loops: {**s**} x {**p**}





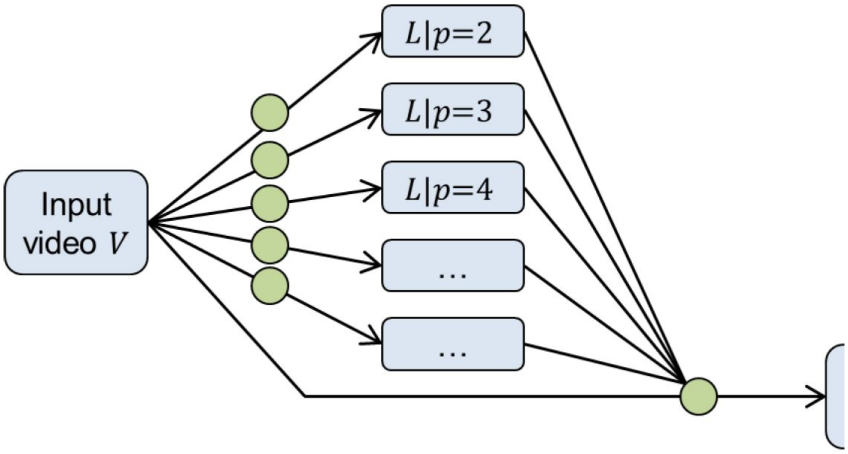
Multilabel Graph Cut on the whole search space doesn’t work

Search algorithm gets stuck in local minima (green = shorter periods):



standard multilabel graph cut | our two-stage approach

Two-stage Approach

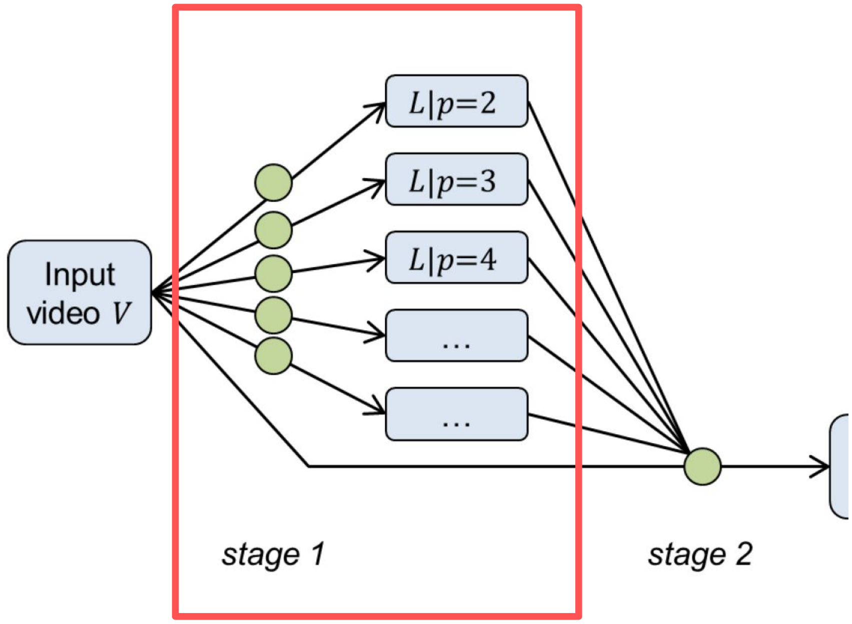


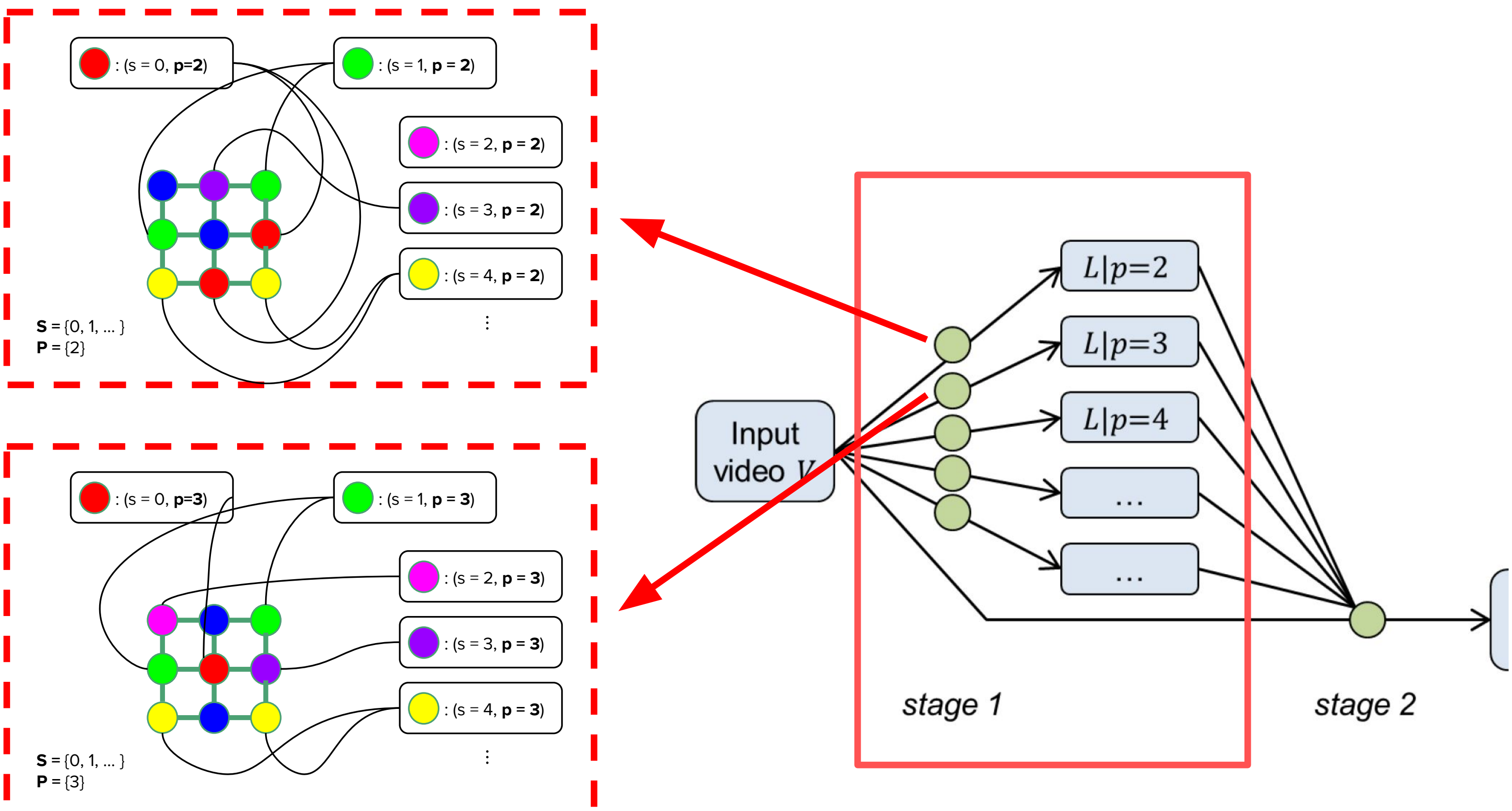
Two-stage Approach

Stage 1: Fix a single loop period for the entire video, and solve for the best start frames

Saves computation cost for spatial consistency

Output: for each period *p*, each pixel has an optimal start frame *s*

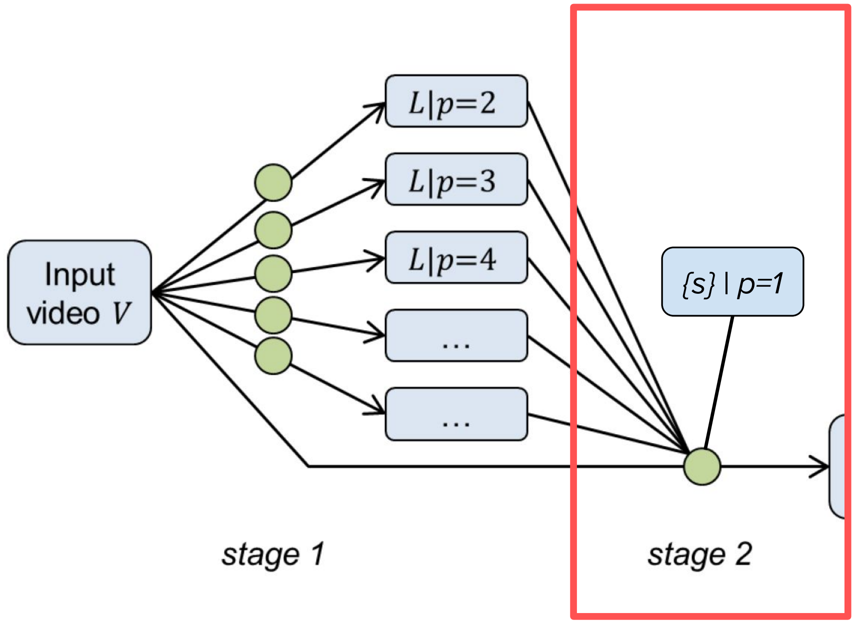




Two-stage Approach

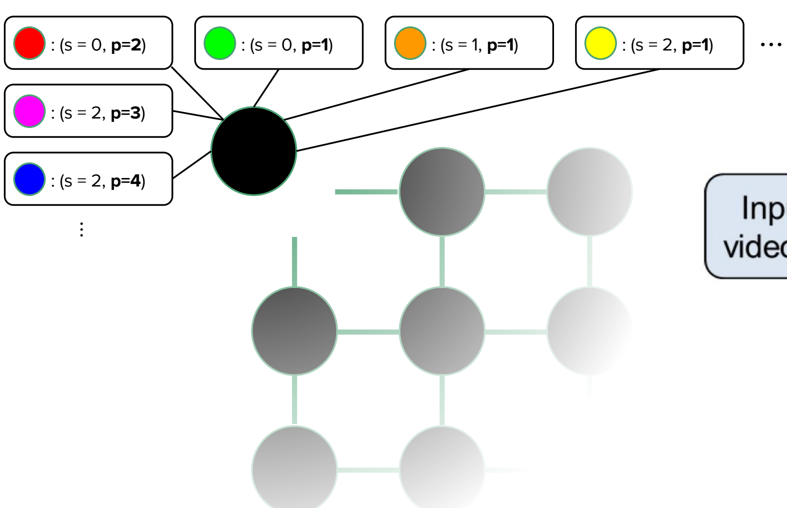
Stage 2: Take optimal start frames from stage 1, and solve for optimal start frame + loop period for each pixel

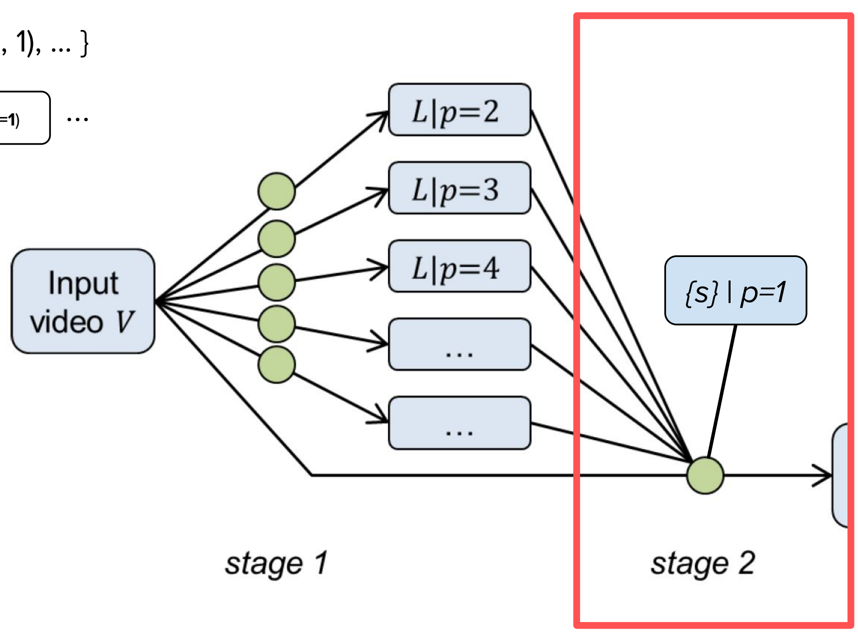
Choices for each pixel:
  | {p} | start frames (stage one)
  | {s} | start frames (p=1)



From stage one:
$\mathbf{s_1} \times \mathbf{p_1} = \{ (0, 2), (2, 3), (2, 4), \ldots \}$

Choices for stage two:
$\mathbf{S_1} \times \mathbf{P_1} = \mathbf{s_1} \times \mathbf{p_1} \cup \{ (0, 1), (1, 1), (2, 1), (3, 1), \ldots \}$
$= \{ (0, 2), (2, 3), (2, 4), \ldots, (0, 1), (1, 1), (2, 1), (3, 1), \ldots \}$



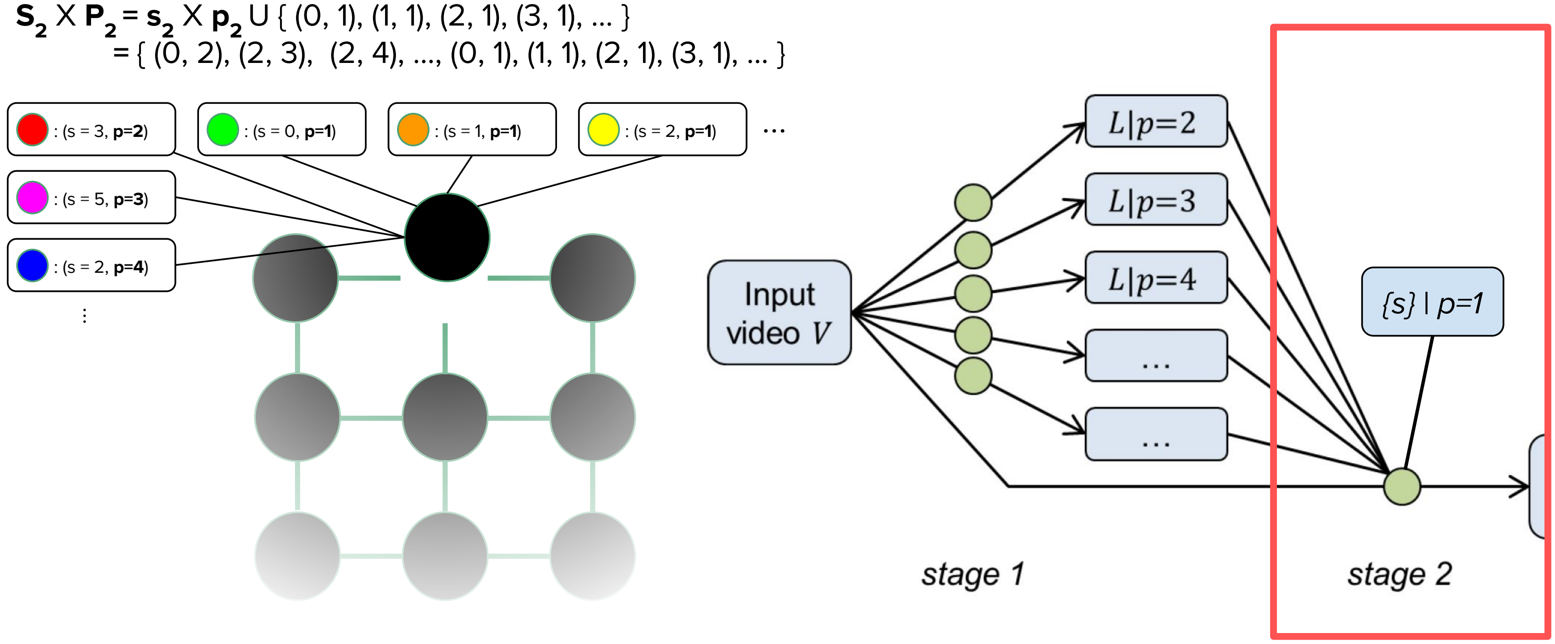


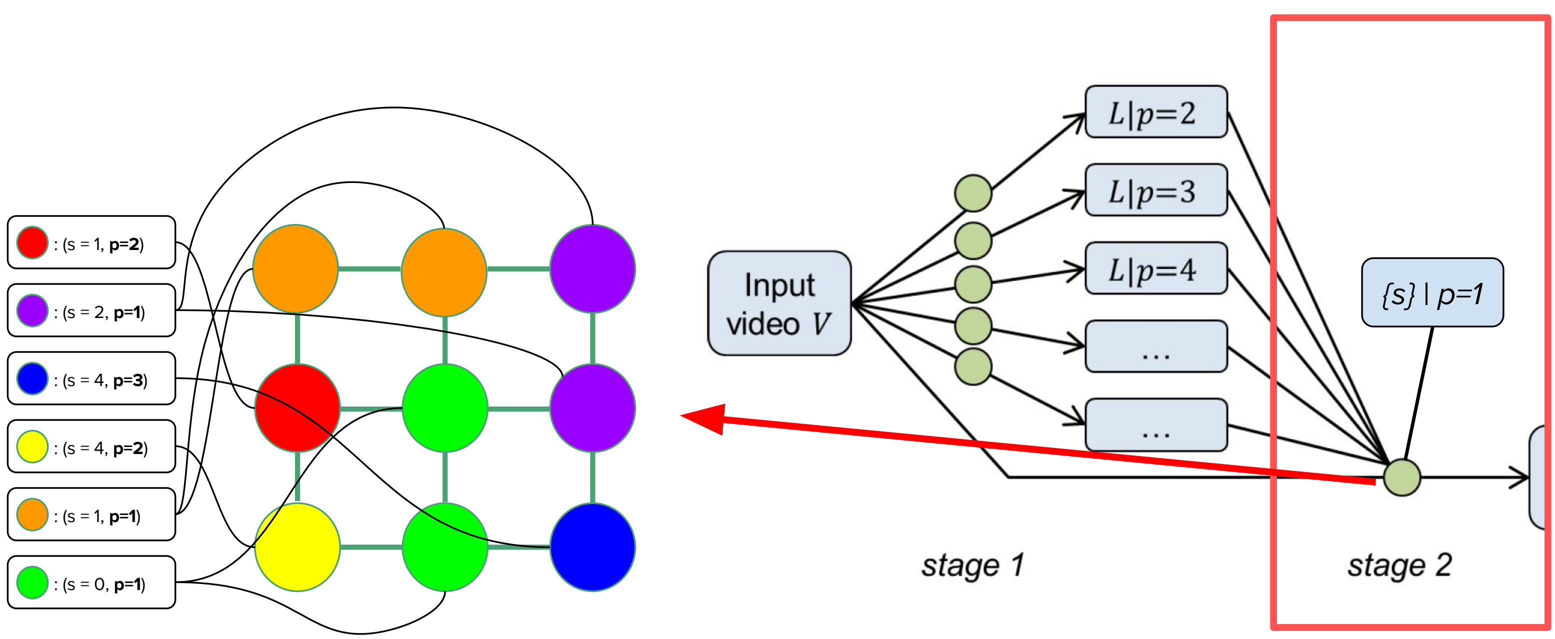
From stage one:

$s_2 \times p_2 = \{ (3, 2), (5, 3), (2, 4), \ldots \}$

Choices for stage two:

$$S_2 \times P_2 = s_2 \times p_2 \cup \{ (0, 1), (1, 1), (2, 1), (3, 1), \ldots \}$$
$$= \{ (0, 2), (2, 3), (2, 4), \ldots, (0, 1), (1, 1), (2, 1), (3, 1), \ldots \}$$





Results

Results

Results

Results

Progressive Video Loops

Example

We Want to Control Dynamism

$$\mathcal{L} = \{L_d \mid 0 \leq d \leq 1\}$$

Pixels: either static or looping

Status: each pixel has an activation threshold *a* (if *d* > *a*, pixel is looping)

Overview

Recall C static: $E_{\text{static}}(x) = c_{\text{static}} \min \left(1, \lambda_{\text{static}} \operatorname{MAD}_{t_i} \| \right.$

1) Solve for most dynamic loop (*d = 1*)
   a) C static to large value: 10
2) Create static loop (*d = 0*)
   a) For each pixel, if static in most dynamic loop, leave as-is
   b) For rest of the pixels, solve for best static frame
3) For each pixel, find activation energy *a*
   a) Recursive binary partition over C static, re-computing *d* every time

Definition of *d*

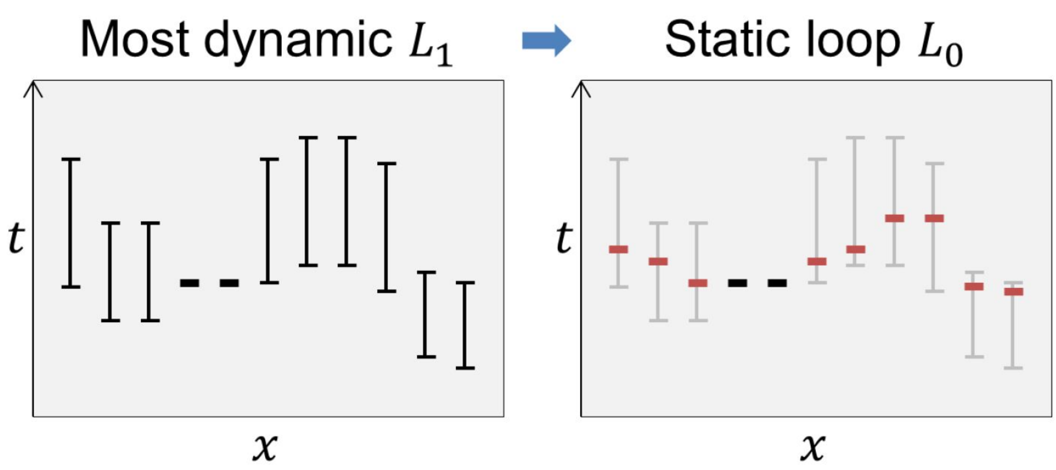
$$\mathrm{Var}(L) = \sum_{x} \mathop{\mathrm{Var}}_{s_x \le t_i < s_x + p_x} \left(V(x, t_i)\right)$$

Temporal Variation of Video Loop

$$\boxed{\mathrm{LOD}(L)} = \mathrm{Var}(L) / \mathrm{Var}(L_1)$$

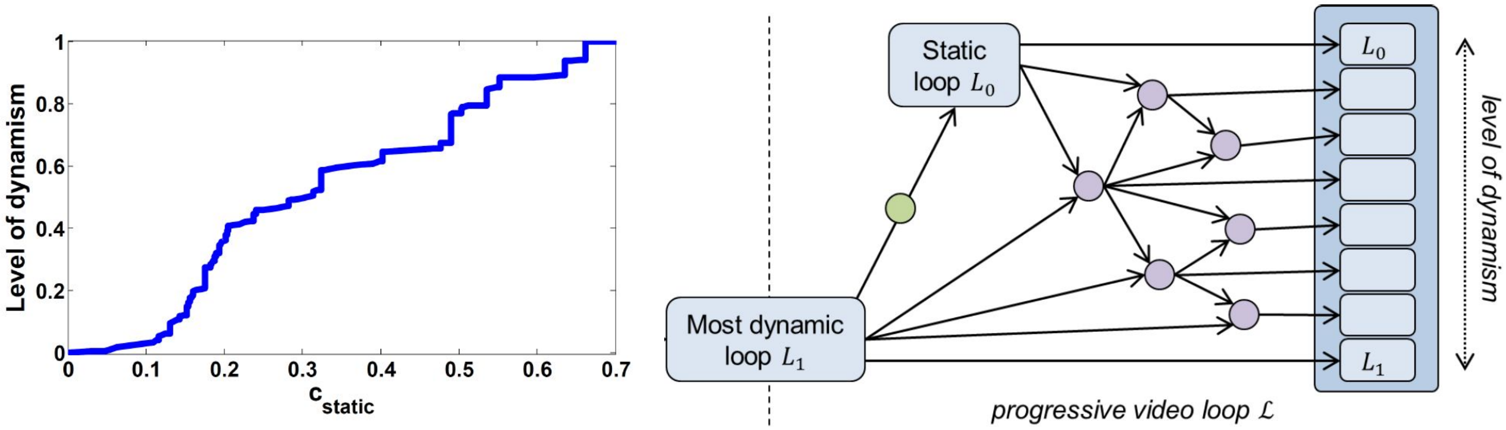
Equation to compute *d* for a Video Loop *L*

Construct Static loop

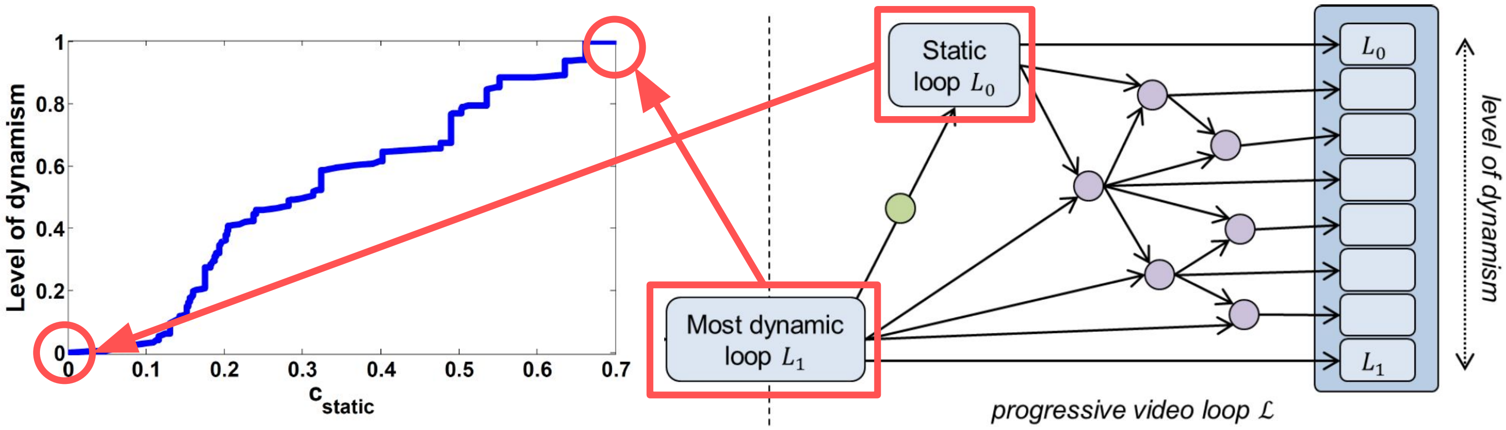


C static = 0; penalize differences between static pixel and median color value

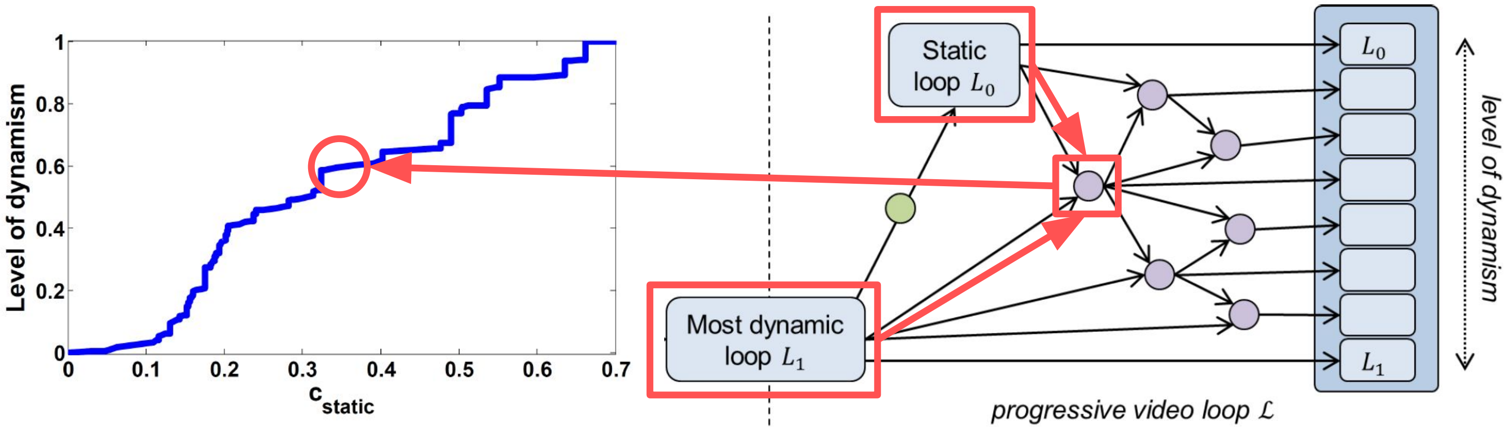
Recursive binary partition



Recursive binary partition

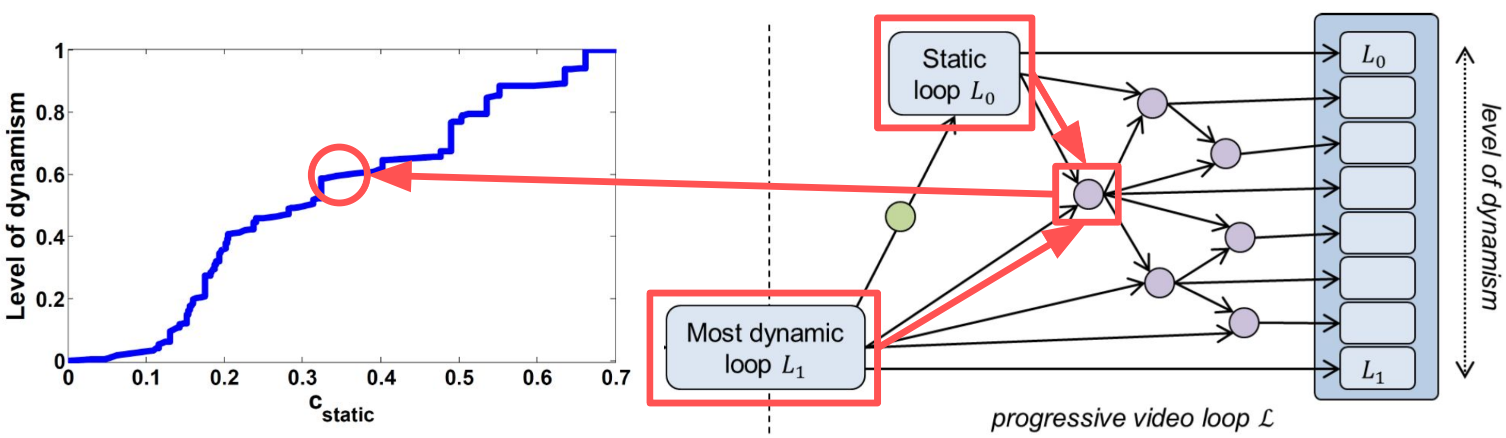


Recursive binary partition

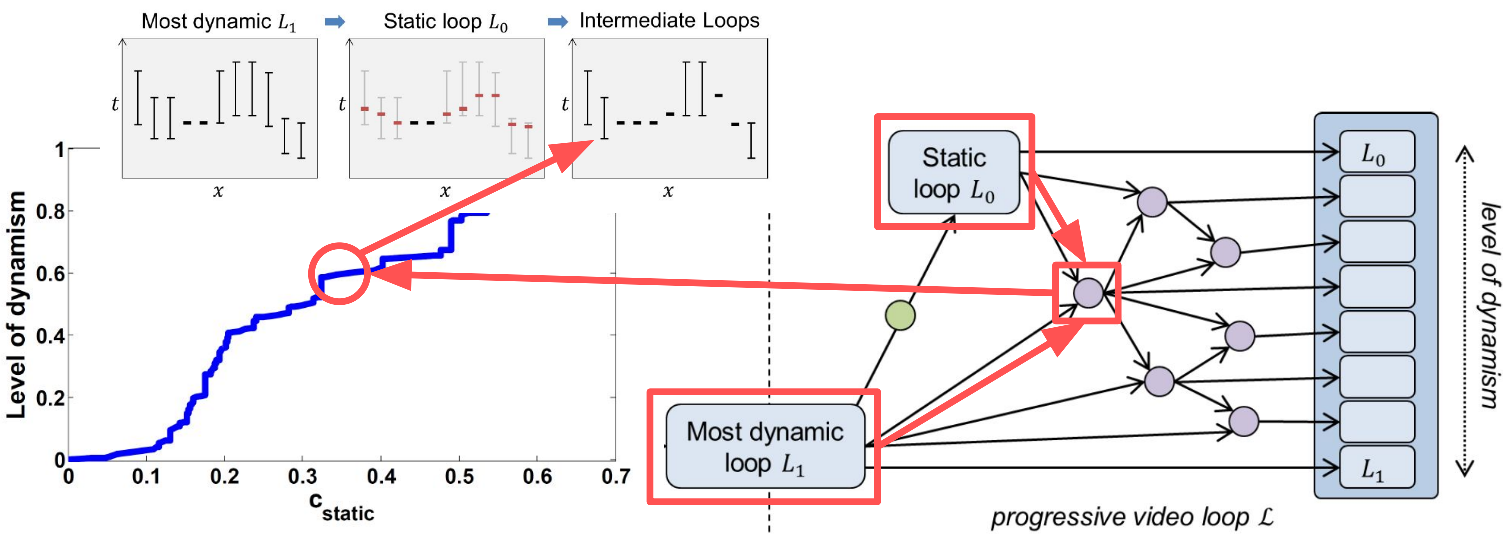


Recursive binary partition

Binary graph cut problem!



Recursive binary partition

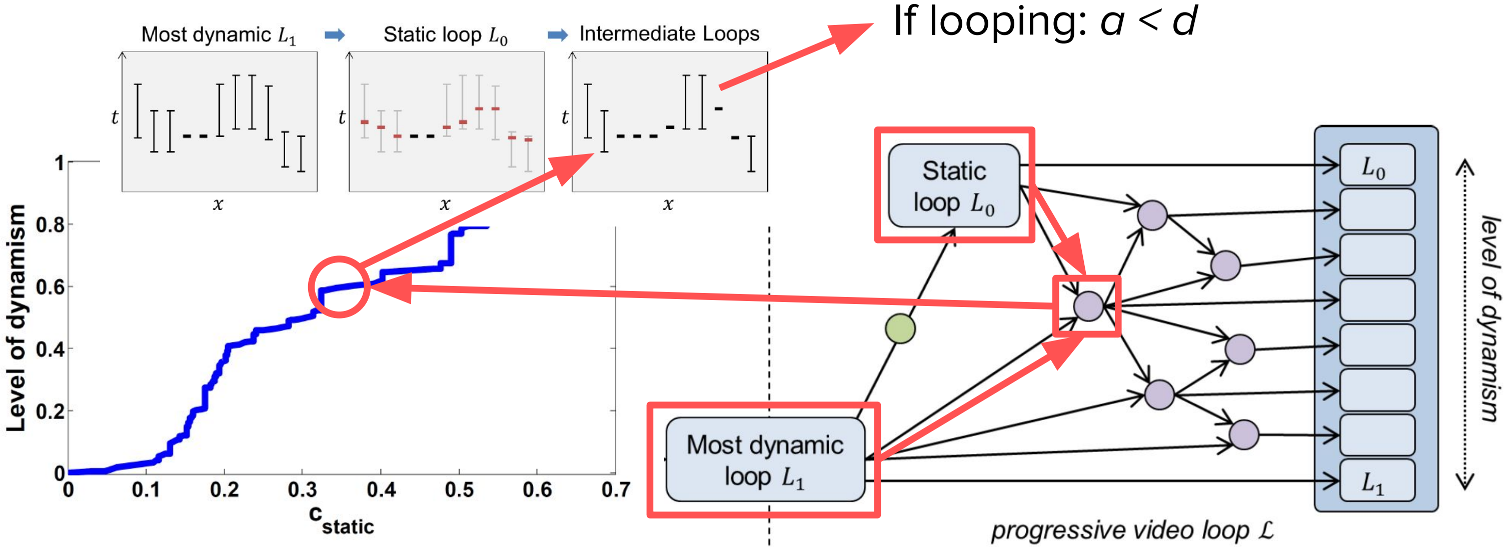


Recursive binary partition

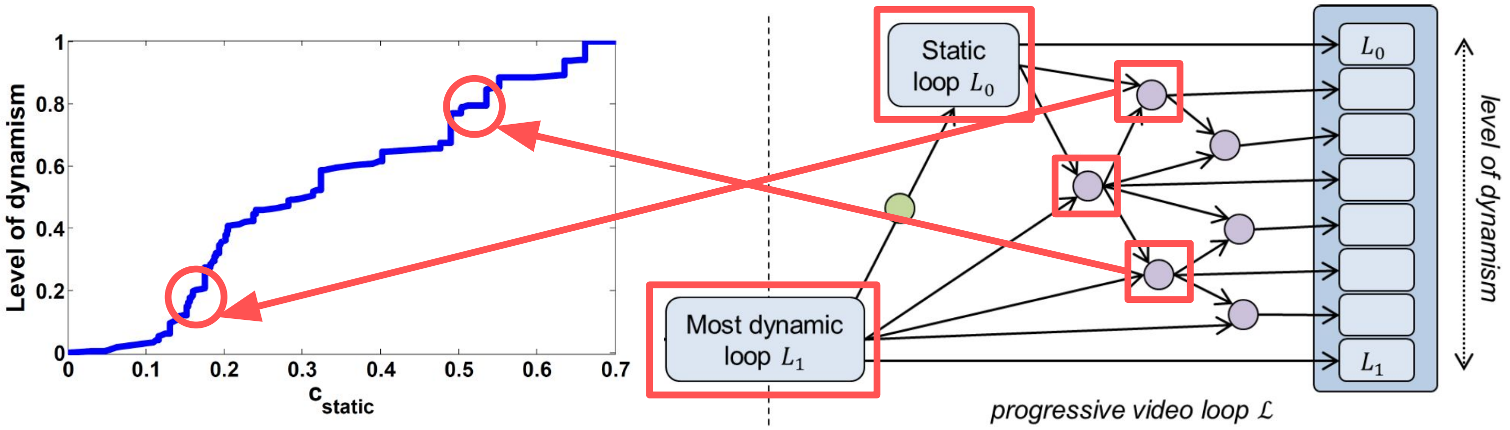
Bounds activation energies:

If static: $a > d$

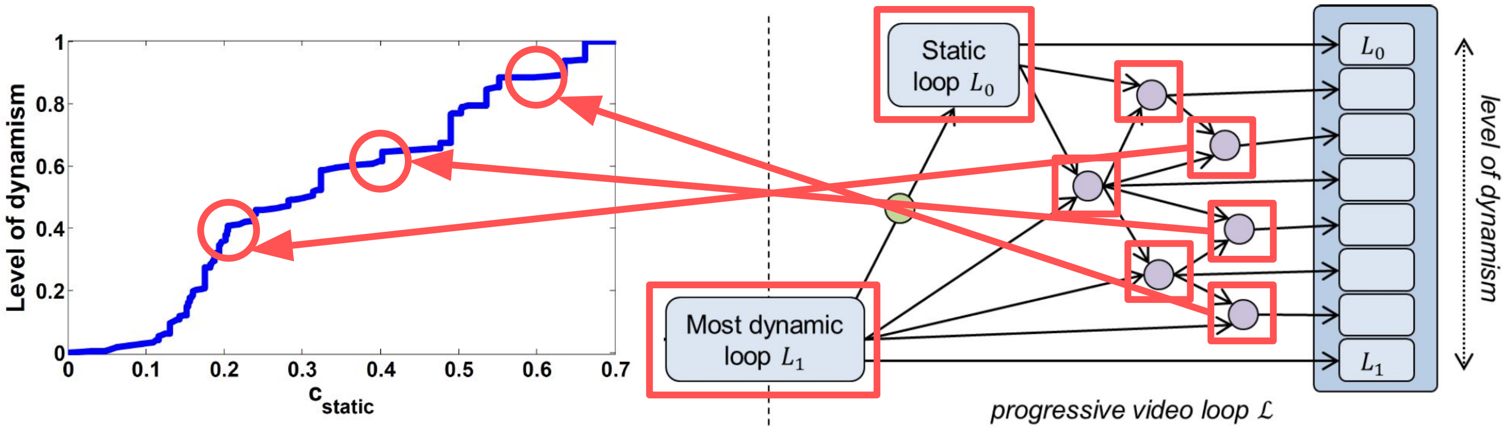
If looping: $a < d$



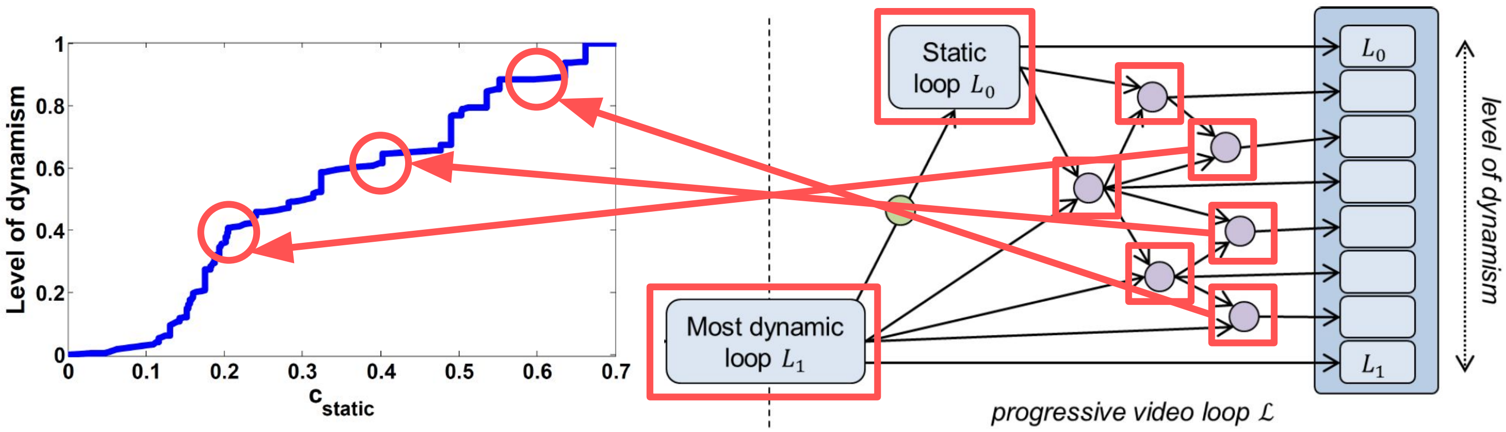
Recursive binary partition



Recursive binary partition



Recursive binary partition



Continue until *d* or C static stop changing
Set activation energies *a* to median of remaining bounds

Ordering of progressive dynamism

$$E_{\text{static}}(x) = c_{\text{static}} \min\left(1, \lambda_{\text{static}} \operatorname*{MAD}_{t_i} \|N(x,t_i) - N(x,t_i+1)\|\right)$$

Re-order which pixels start looping first

$$E_{\text{static}}(x) =$$
$$c_{\text{static}}\left(1.05 - \min\left(1, \lambda_{\text{static}} \operatorname*{MAD}_{t_i} \|N(x,t_i) - N(x,t_i+1)\|\right)\right)$$

Result

Results: Limitations

Original

Looping

Thank you for listening!