國立高雄科技大學
National Kaohsiung University of Science and Technology

Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images

Yuri Y Boykov and Marie-Pierre Jolly

IEEE Intl. Conf. on Computer Vision (ICCV), 2001

Speaker: Shih-Shinh Huang

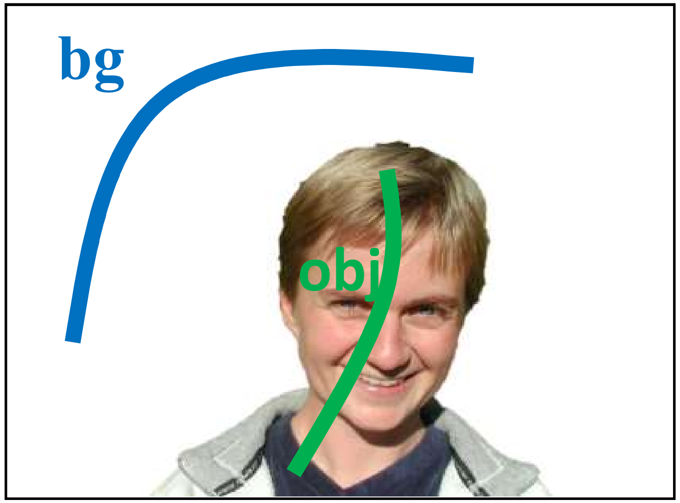
July 10, 2024

Outline

- Introduction
  - Motivation
  - Problem Description
  - Central Idea
- Graph-Based Modeling
  - Graph Construction
  - Cost Function Definition
  - Cost Minimization

Introduction

- Motivation
  - The problems from automatic segmentation can be alleviated via user guidance

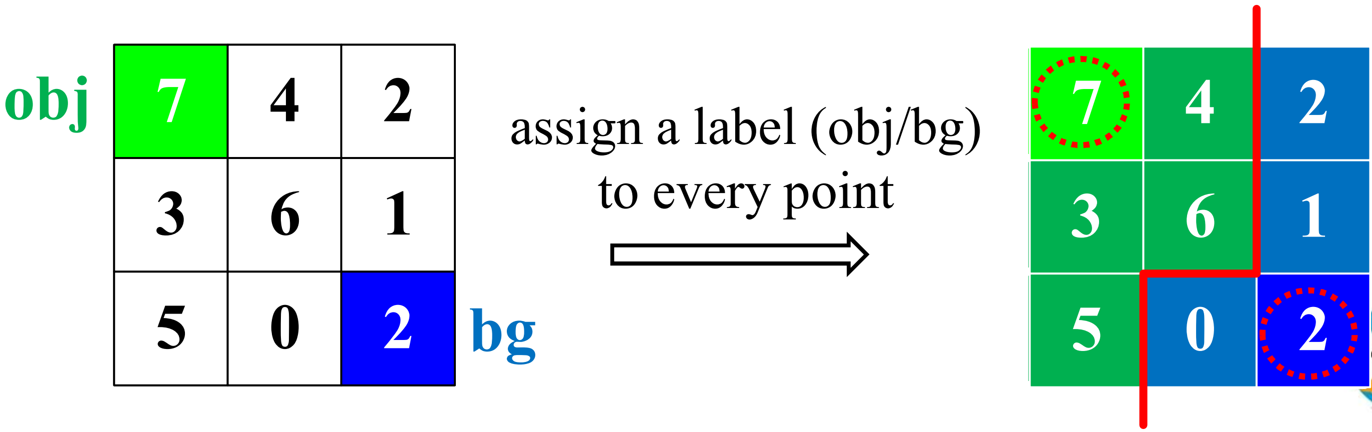


interactive segmentation

- identify the parts of the object (**obj**) and background (**bg**)
- compute a segmentation that adheres to the user setting

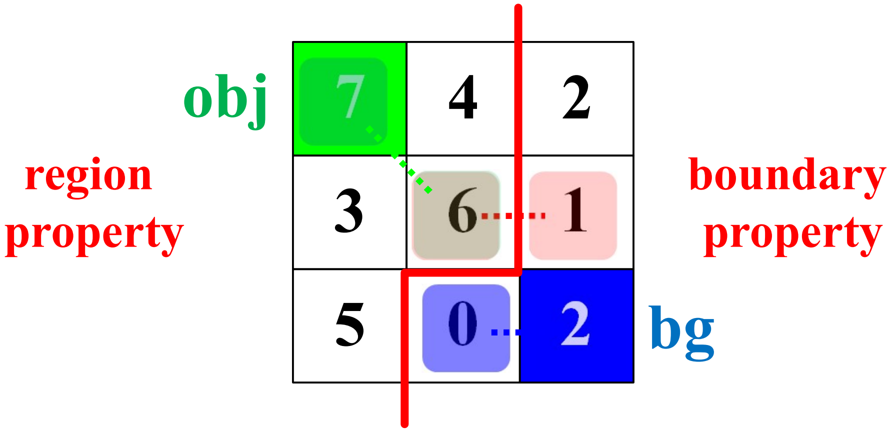
Introduction

- Problem Description
  - Given: an image and two sets of **obj/bg** points provided by the user
  - Goal: find a segmentation (an assignment of points)
    - well separate the object points from background ones
    - adhere to user's constraints



Introduction

- Central Idea
  - enforce both **boundary** and **region** properties
    - boundary property: pixels with significant intensity differences from their neighbors are boundaries
    - region property: pixels with similar intensities to **obj/bg** are inclined to be labeled as **obj/bg**



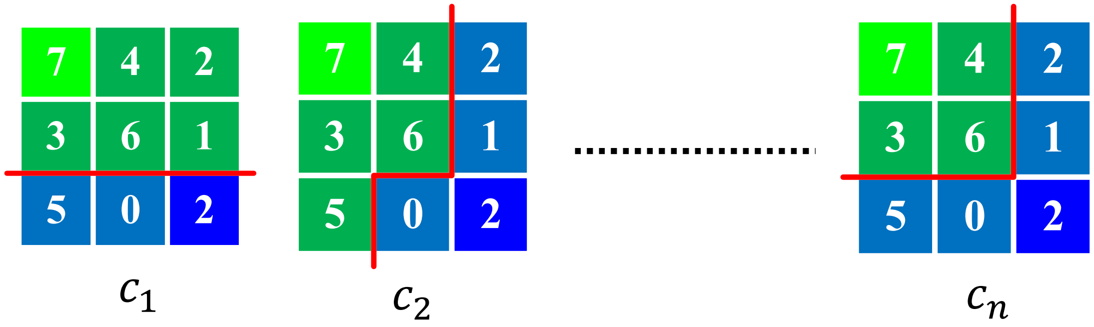
Introduction

- Central Idea
  - formulate segmentation as an **optimization** problem
    - define a cost function $E(.)$ based on two properties of the segmentation c.

$$E(c) = \lambda R(c) + B(c)$$

a segmentation — region / importance — boundary

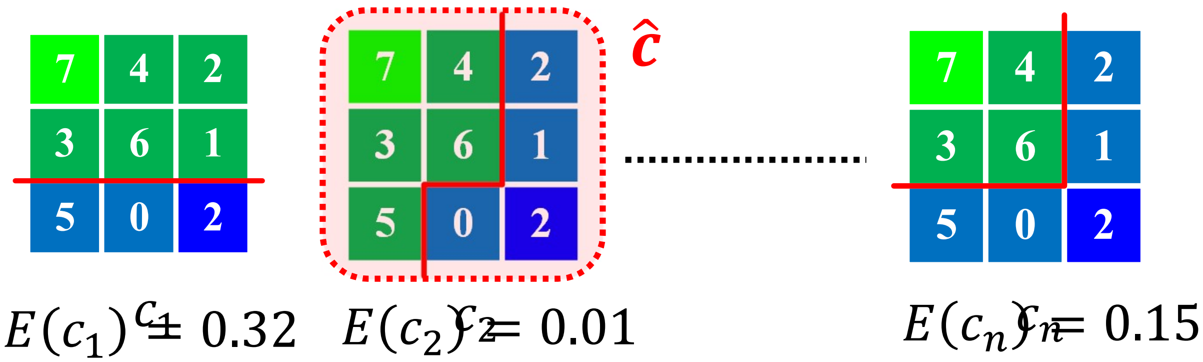
region cost function — boundary cost function



Introduction

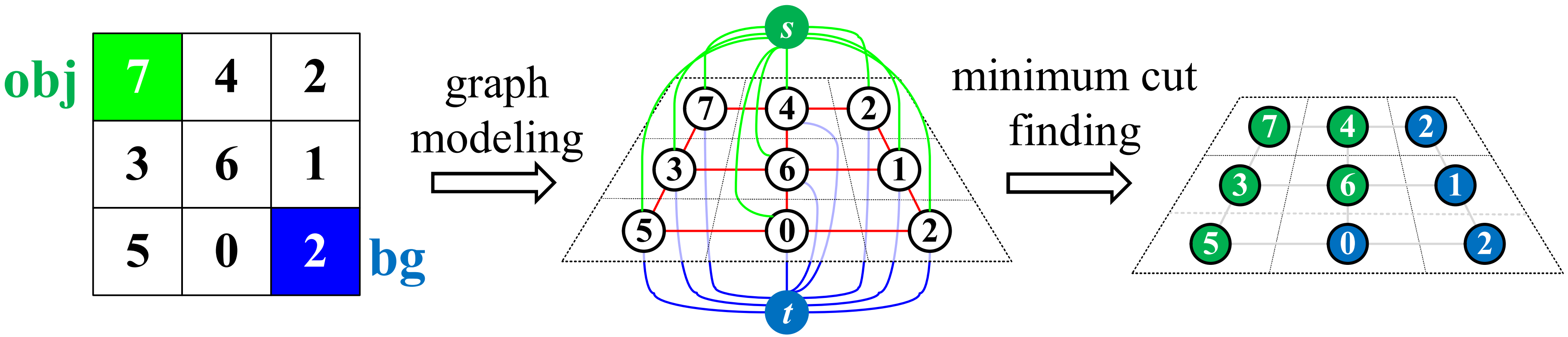
- Central Idea
  - formulate segmentation as an **optimization** problem
    - minimize $E(.)$ to find an optimal segmentation $\hat{\boldsymbol{c}}$

$$\hat{\boldsymbol{c}} = \underset{c}{\operatorname{argmin}}\, E(\boldsymbol{c})$$



Introduction

- Central Idea
  - solve the problem via the **graph-cut** algorithm
    - model the optimization problem as a **graph**
    - determine the $\hat{c}$ through **minimum cut** algorithm



Graph-Based Modeling

- Graph Construction
  - use an undirected graph $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{W})$ to describe an image
    - $\mathbf{V} = \{\boldsymbol{v}\}$: a set of vertices
    - $\mathbf{E} = \{\boldsymbol{e}\}$: a set of edges $\boldsymbol{e} = (\boldsymbol{v_i}, \boldsymbol{v_j})$ defined under a neighborbooh system.
    - $\mathbf{W} = \{\boldsymbol{w}\}$: a set of weights assigned to the edges $\mathbf{E}$

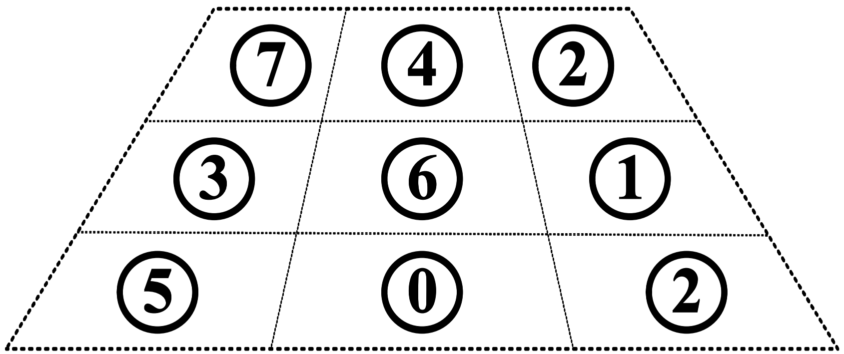
Graph-Based Modeling

- Graph Construction: **V** (vertices) definition
  - model each pixel as a pixel vertex $\boldsymbol{v}$
  - add two terminal vertices **source** ($\boldsymbol{s}$) and **sink** ($\boldsymbol{t}$)

$$\mathbf{V} = \{\boldsymbol{v}\} \cup \{\boldsymbol{s}, \boldsymbol{t}\}$$

s source (object)

7	4	2
3	6	1
5	0	2



sink (background) *t*

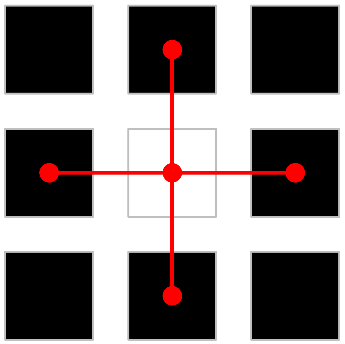
Graph-Based Modeling

- Graph Construction: **E** (edges) definition
  - model spatial relation between $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$ by a **neighboring edge** $\boldsymbol{e_{i,j}} = (\boldsymbol{v_i}, \boldsymbol{v_j})$
  - add two **terminal edges** $\boldsymbol{e_{s,i}} = (\boldsymbol{s}, \boldsymbol{v_i})$ and $\boldsymbol{e_{t,i}} = (\boldsymbol{t}, \boldsymbol{v_i})$ for every pixel vertex $\boldsymbol{v_i}$

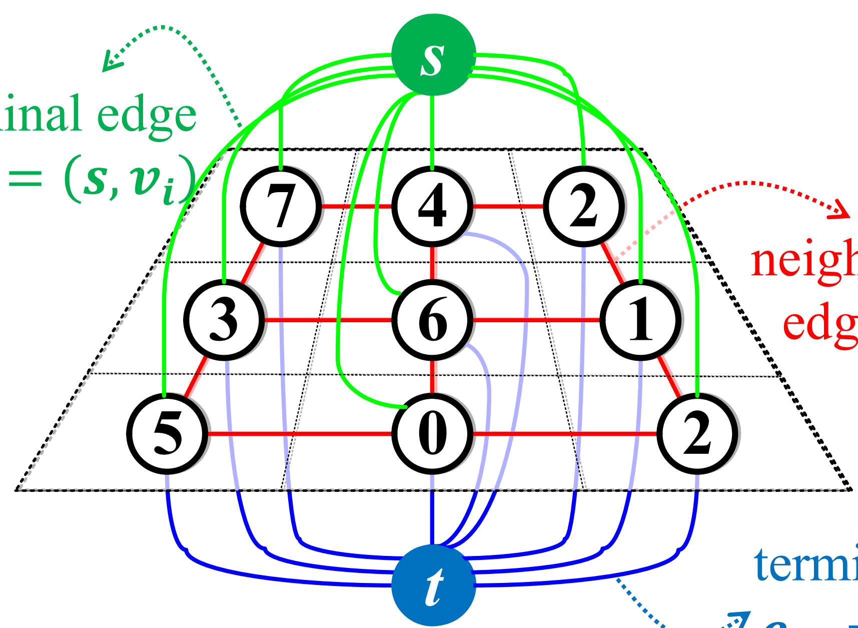
$$\mathbf{E} = \{\boldsymbol{e_{i,j}}\} \cup \{\boldsymbol{e_{s,i}}\} \cup \{\boldsymbol{e_{t,j}}\}$$

Graph-Based Modeling

- Graph Construction: **E** (edges) definition

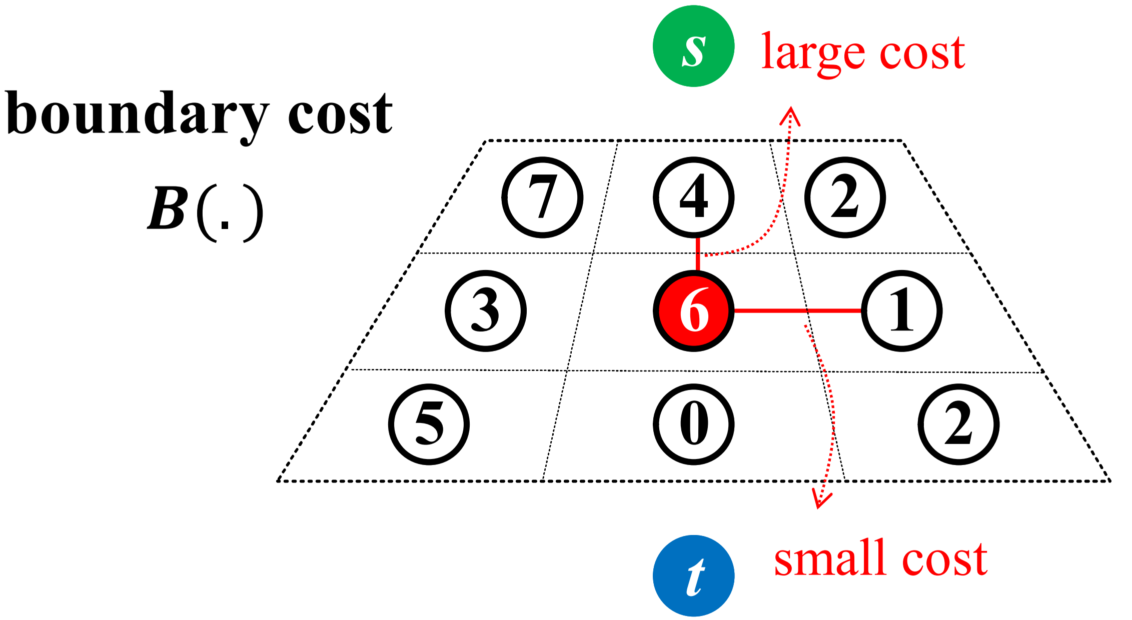


4-connectivity (N_4)



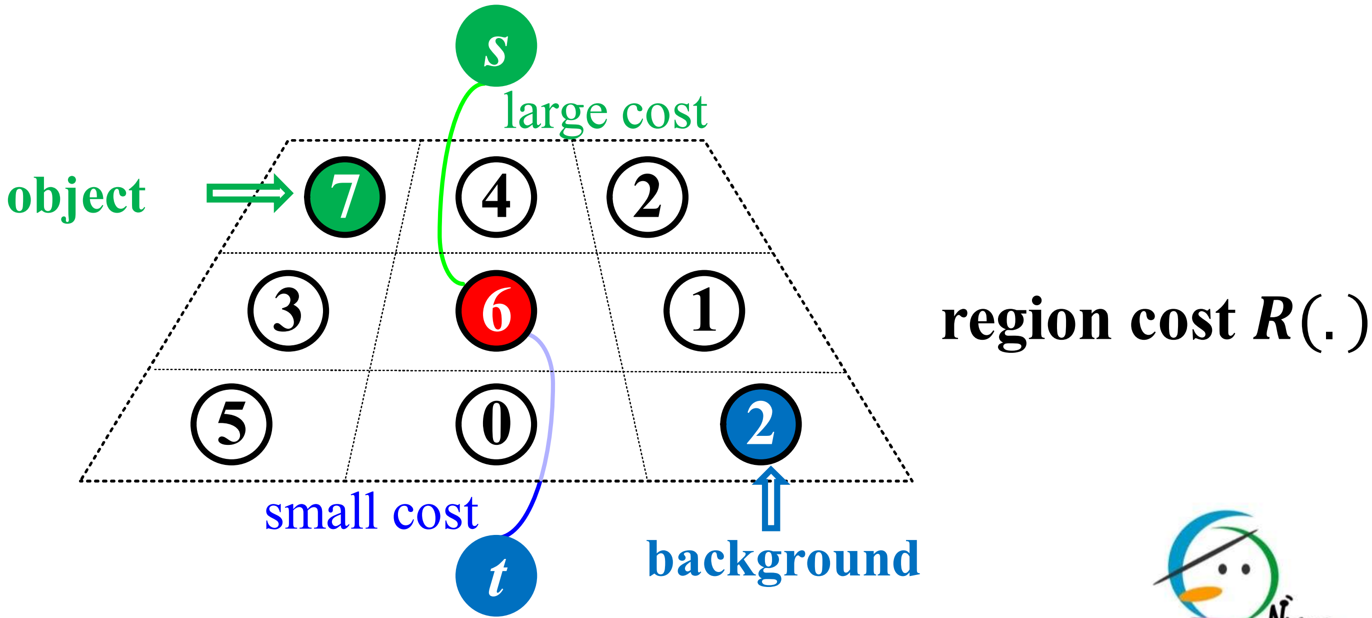
Graph-Based Modeling

- Graph Construction: **W** (weights) definition
  - a weight assigned to an edge $\boldsymbol{e_{ij}} = (\boldsymbol{v_i}, \boldsymbol{v_j})$ is the **penalty** (cost) when $\boldsymbol{l(v_i)} \neq \boldsymbol{l(v_j)}$



Graph-Based Modeling

- Graph Construction: **W** (weights) definition
  - a weight assigned to an edge $\boldsymbol{e_{ij}} = (\boldsymbol{v_i}, \boldsymbol{v_j})$ is the **<u>penalty</u>** (cost) when $\boldsymbol{l(v_i)} \neq \boldsymbol{l(v_j)}$



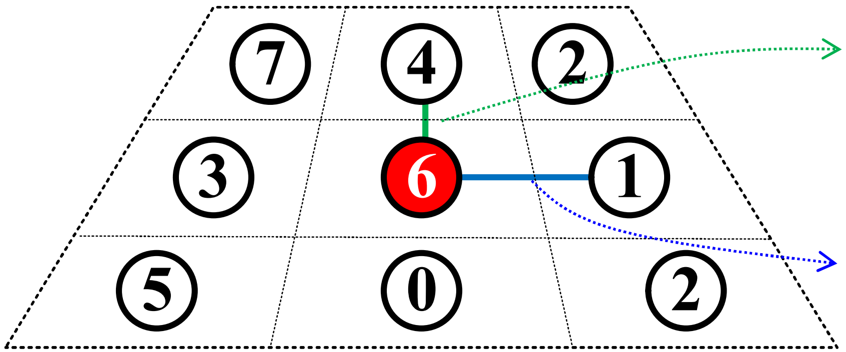
Graph-Based Modeling

- Cost Function Definition: $\boldsymbol{e_{i,j}} = (\boldsymbol{v_i}, \boldsymbol{v_j})$
  - $\boldsymbol{w}(\boldsymbol{e_{i,j}})$ is modeled using **<u>ad-hoc</u>** function

$$\boldsymbol{w}(\boldsymbol{e_{i,j}}) = exp\left(-\frac{\left(I_{v_i} - I_{v_j}\right)^2}{2 \times \sigma^2}\right)$$

pixel intensity

camera noise (2.0)

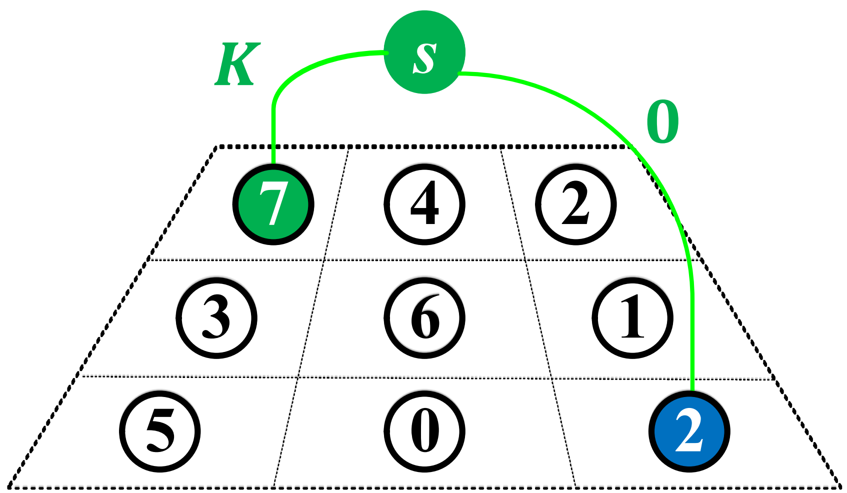


$$\exp\left(-\frac{(6-4)^2}{2 \times 2^2}\right) = \exp(-0.5) = 0.60$$

$$\exp\left(-\frac{(6-1)^2}{2 \times 2^2}\right) = \exp(-3.125) = 0.04$$

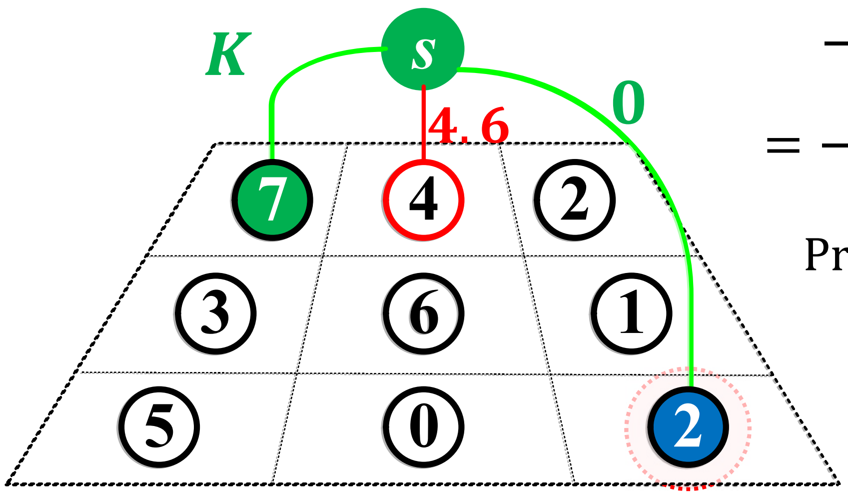
Graph-Based Modeling

- Cost Function Definition: $\boldsymbol{e_{s,i}} = (\boldsymbol{s}, \boldsymbol{v_i})$
  - $\boldsymbol{w(e_{s,i})}$ is the **<u>penalty</u>** (cost) when $\boldsymbol{l(v_i)} \neq$ **<u>obj</u>**
    - $\boldsymbol{v_i}$ is **obj** vertex: $\boldsymbol{w(e_{s,i})} = \boldsymbol{K}$ (large weight)
    - $\boldsymbol{v_i}$ is **bg** vertex: $\boldsymbol{w(e_{s,i})} = 0$



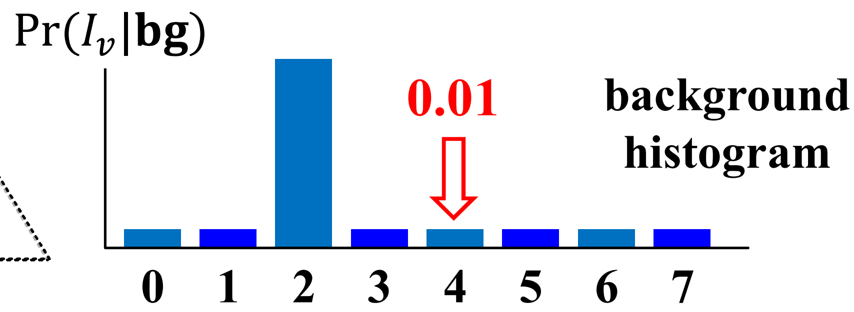
Graph-Based Modeling

- Cost Function Definition: $\boldsymbol{e_{s,i} = (s, v_i)}$
  - $\boldsymbol{w(e_{s,i})}$ is the **penalty** (cost) when $\boldsymbol{l(v_i)} \neq$ **obj**
    - $\boldsymbol{v_i}$ is **UNK** vertex: **negative log-likelihood** of intensity distribution of **background region**



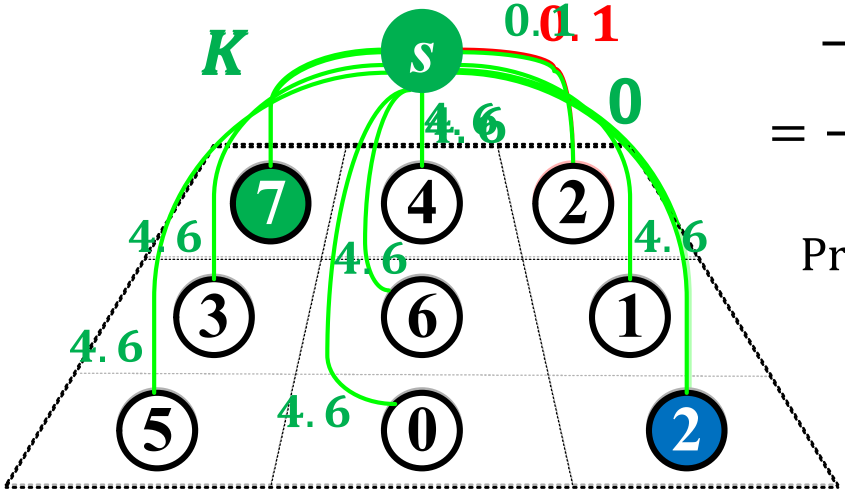
$$-\ln(\Pr(I_v = 4|\mathbf{bg})) \times \lambda$$

$$= -\ln(0.01) \times 1 = 4.60$$

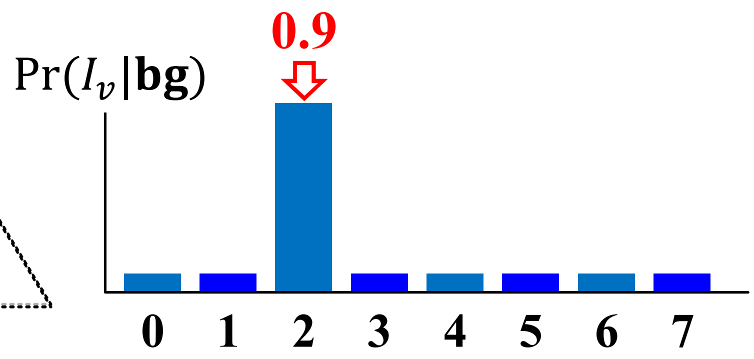


Graph-Based Modeling

- Cost Function Definition: $\boldsymbol{e_{s,i} = (s, v_i)}$
  - $\boldsymbol{w(e_{s,i})}$ is the **penalty** (cost) when $\boldsymbol{l(v_i) \neq}$ **obj**
    - $\boldsymbol{v}$ is **UNK** vertex: **negative log-likelihood** of intensity distribution of **background region**

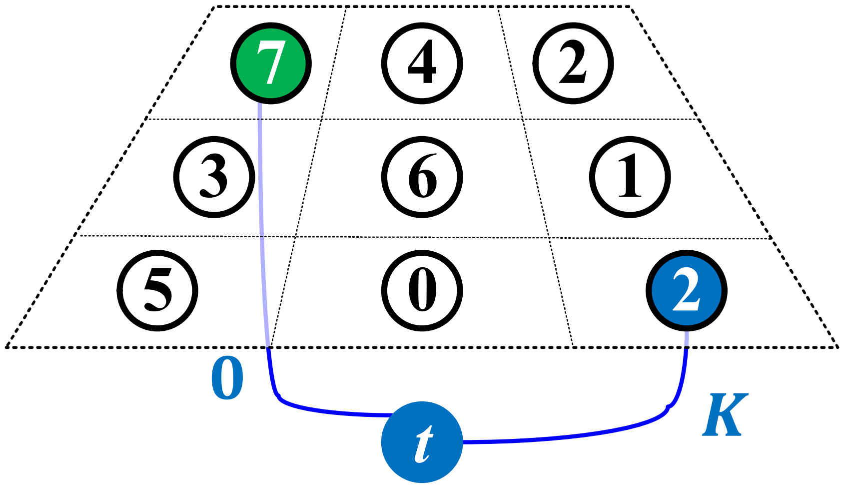


$$-\ln(\Pr(I_v = 2|\mathbf{bg})) \times \lambda$$
$$= -\ln(0.9) \times 1 = 0.10$$



Graph-Based Modeling

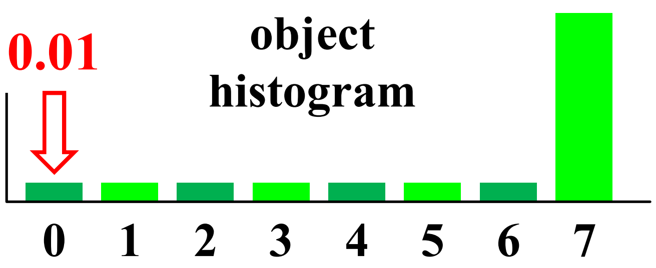
- Cost Function Definition: $\boldsymbol{e_{t,i}} = (\boldsymbol{t}, \boldsymbol{v_i})$
  - $\boldsymbol{w(e_{t,i})}$ is the **penalty** (cost) when $\boldsymbol{l(v_i)} \neq$ **bg**
    - $\boldsymbol{v_i}$ is **obj** vertex: $\boldsymbol{w(e_{t,i}) = 0}$
    - $\boldsymbol{v_i}$ is **bg** vertex: $\boldsymbol{w(e_{t,i}) = K}$ (large weight)

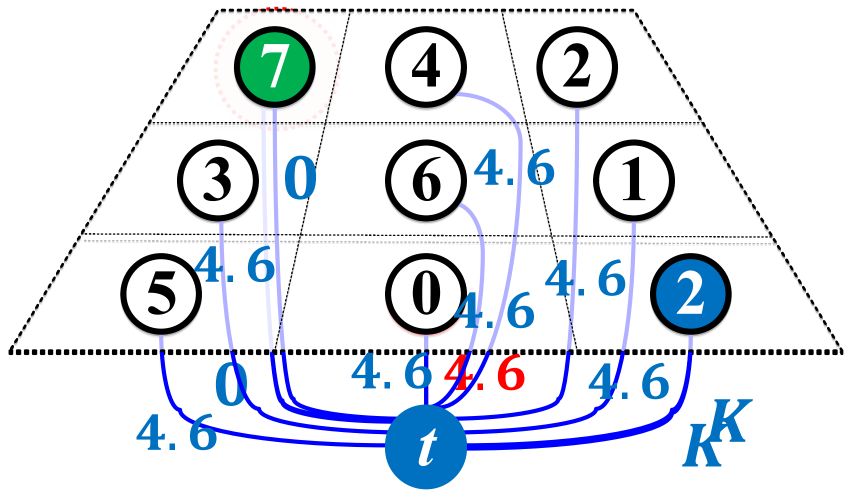


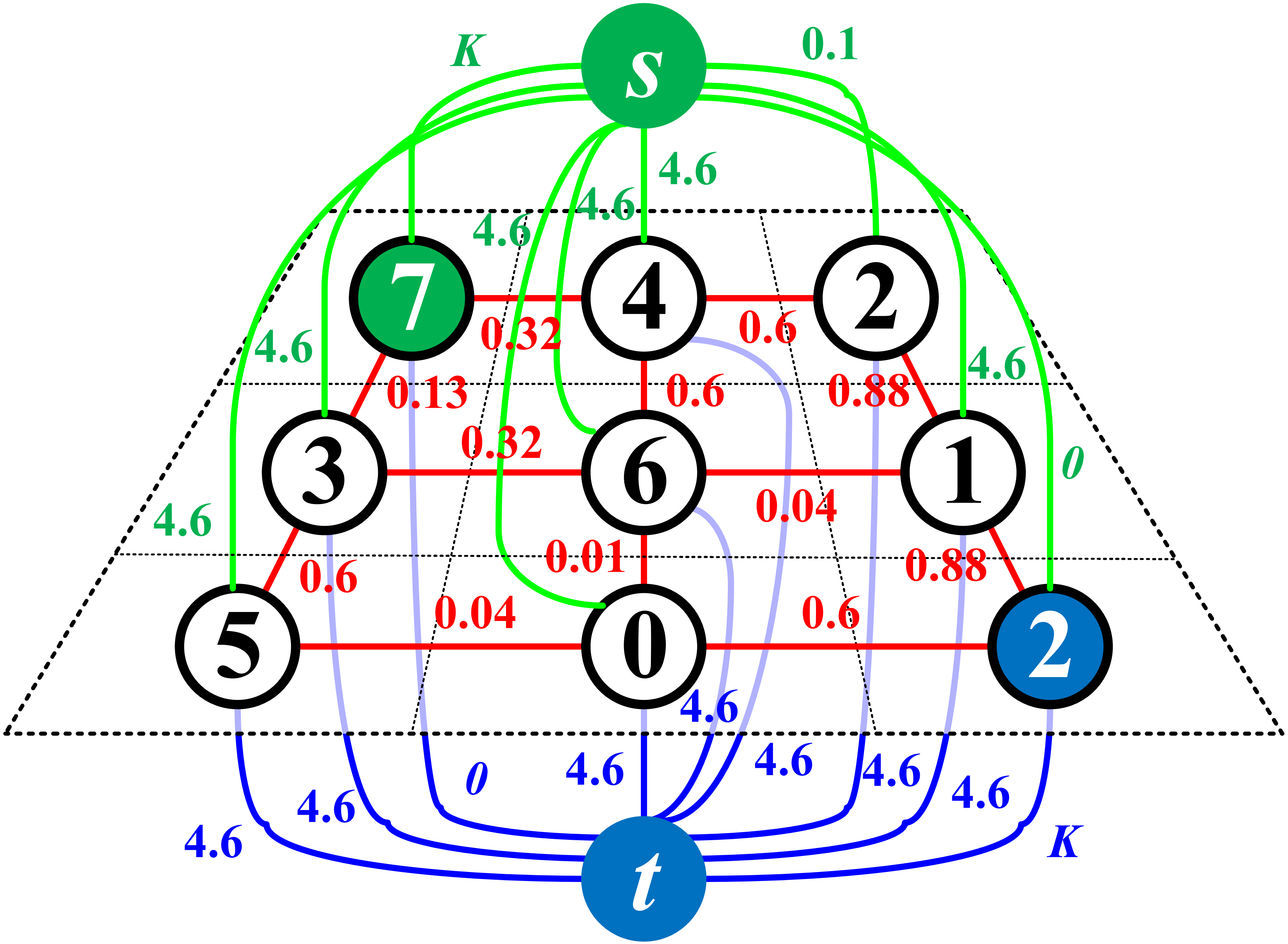
Graph-Based Modeling

- Cost Function Definition: $\boldsymbol{e_{t,i} = (t, v_i)}$
  - $\boldsymbol{w(e_{t,i})}$ is the **penalty** (cost) when $\boldsymbol{l(v_i) \neq}$ **bg**
    - $\boldsymbol{v_i}$ is **UNK** vertex: **negative log-likelihood** of intensity distribution of **object region**

$$-\ln(\Pr(I_v = 0|\mathbf{obj})) \times \lambda$$
$$= -\ln(0.01) \times 1 = 4.6$$







s
t
K
0.1
4.6
4.6
4.6
4.6
4.6
4.6
0
7
4
2
3
6
1
5
0
2
0.32
0.6
0.13
0.6
0.88
0.32
0.04
0.01
0.6
0.88
0.04
0.6
4.6
0
4.6
4.6
4.6
4.6
4.6
4.6
4.6
K

Graph-Based Modeling

- Cost Minimization
  - A segmentation is a **partition** (**cut**) $\boldsymbol{c} = (\mathbf{S}, \mathbf{T})$ of vertices into two disjoint sets **S** and **T**
    - $\mathbf{S} \cup \mathbf{T} = \mathbf{V}$ and $\mathbf{S} \cap \mathbf{T} = \emptyset$
    - $\boldsymbol{s} \in \mathbf{S}$ and $\boldsymbol{t} \in \mathbf{T}$
  - The cost of a partition (cut) $\boldsymbol{c}$ is the sum of the weights of edges between **S** and **T**

$$\boldsymbol{E}(\boldsymbol{c}) = \sum_{e=(v_i \in \mathbf{S}, v_j \in \mathbf{T})} \boldsymbol{w}(\boldsymbol{e})$$

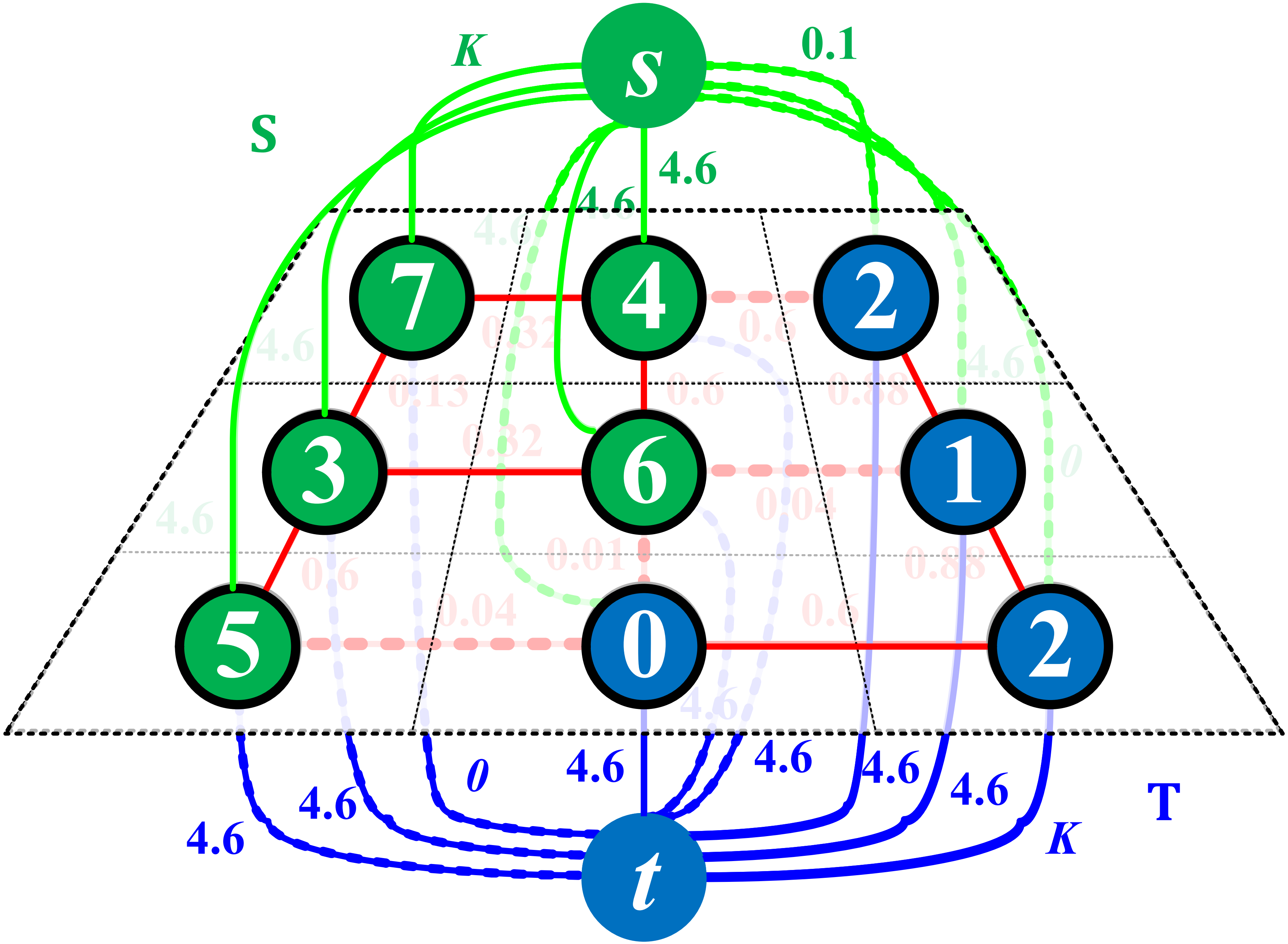
Graph-Based Modeling

- Cost Minimization
  - The optimal segmentation $\hat{\boldsymbol{c}}$ is the cut with the minimum cost (**minimum cut**).

$$\hat{\boldsymbol{c}} = \underset{c}{\operatorname{argmin}}\, E(\boldsymbol{c})$$

  - The minimum cut can be found via well-known **maximum flow** algorithm

Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," *EMMCVPR*, 2001



s
K
0.1
S
4.6
4.6
7
4
2
3
6
1
5
0
2
0.32
0.6
0.13
0.6
0.88
0.32
0.04
0
0.01
0.88
0.6
0.04
0.6
4.6
4.6
4.6
4.6
4.6
4.6
4.6
4.6
0
4.6
4.6
4.6
4.6
4.6
T
K
t


thank you
© 1996, 2002 SANRIO CO., LTD.