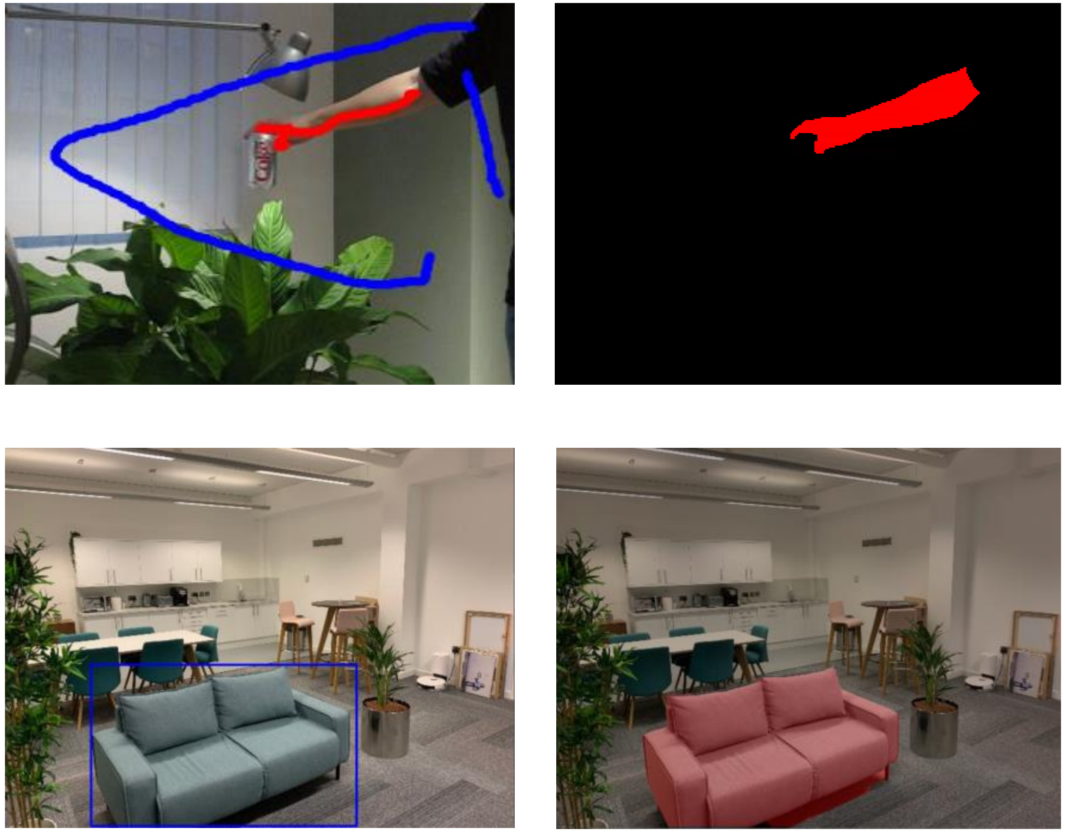
COMP2005

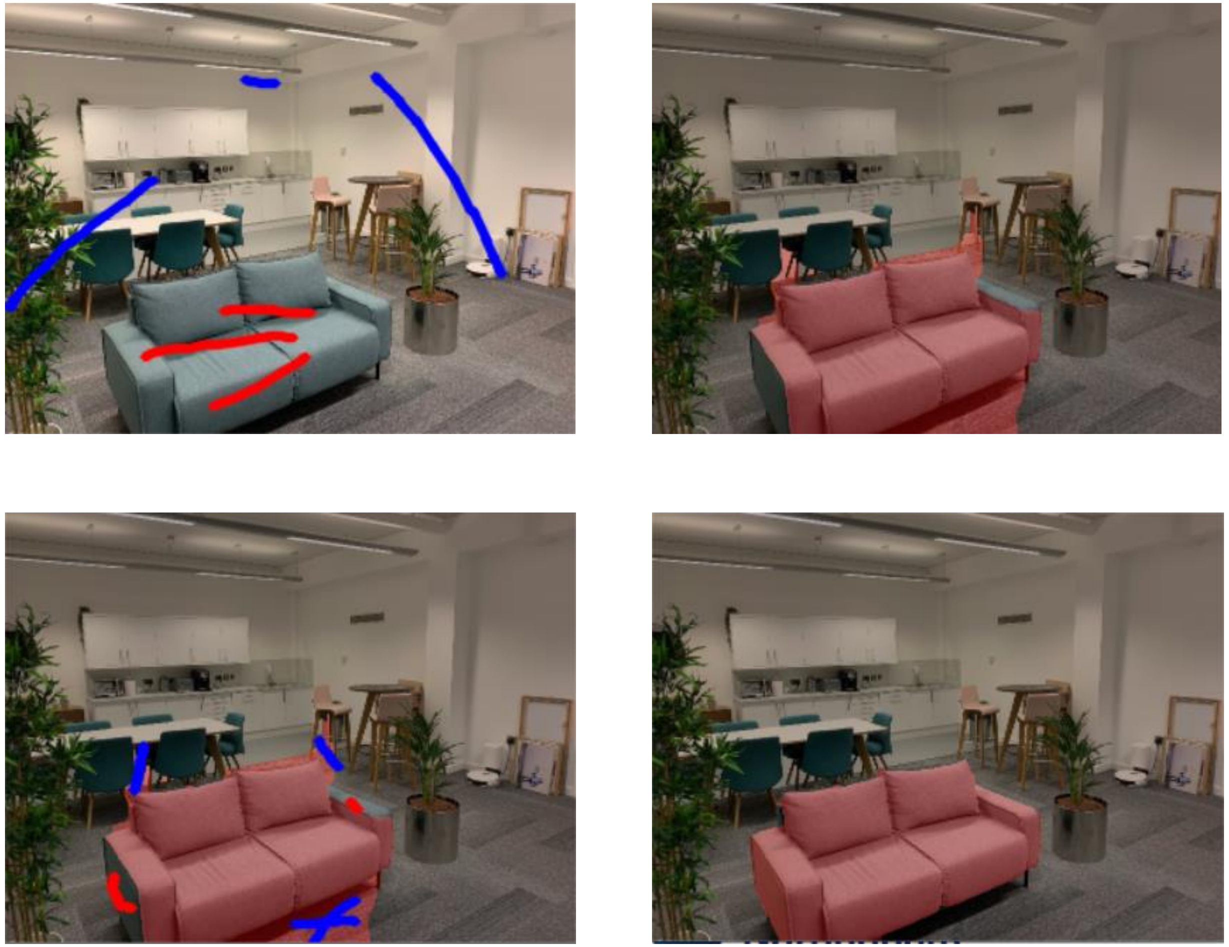
Interactive Segmentation: Graph Cut

What is Interactive Segmentation?

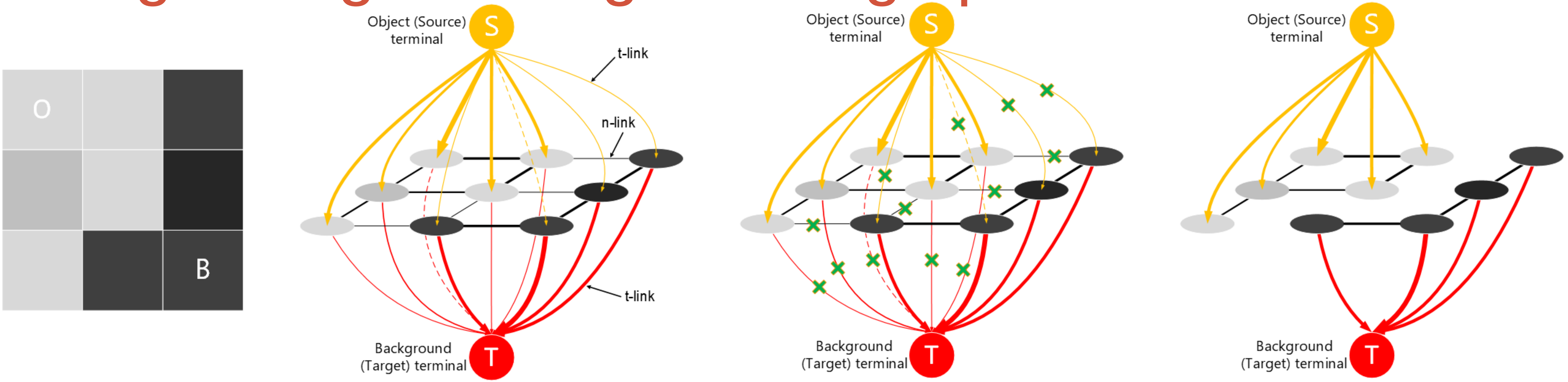


- The user selcect a rough area in the image to indicate the content she/he would like to segment
- The algorithm segments the area of the user’s interest
- We will focus on Graph Cut in this module

Graph Cut: Overview



Organizing the image into a graph

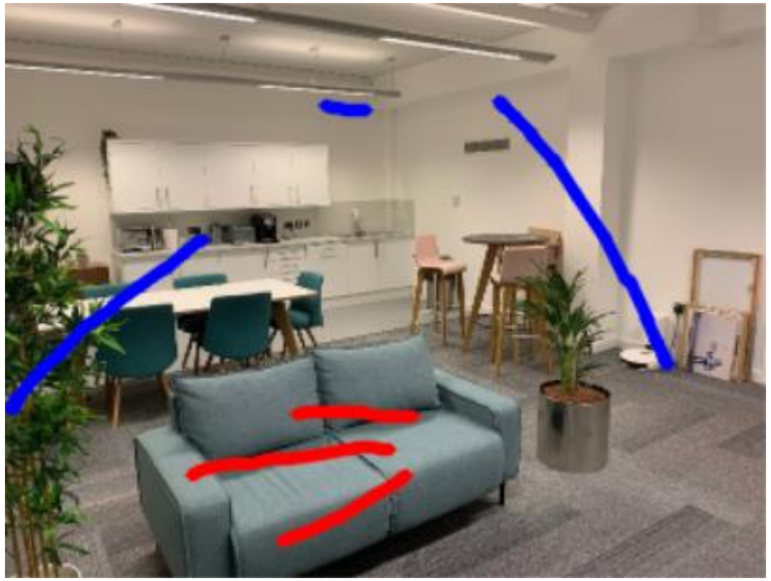


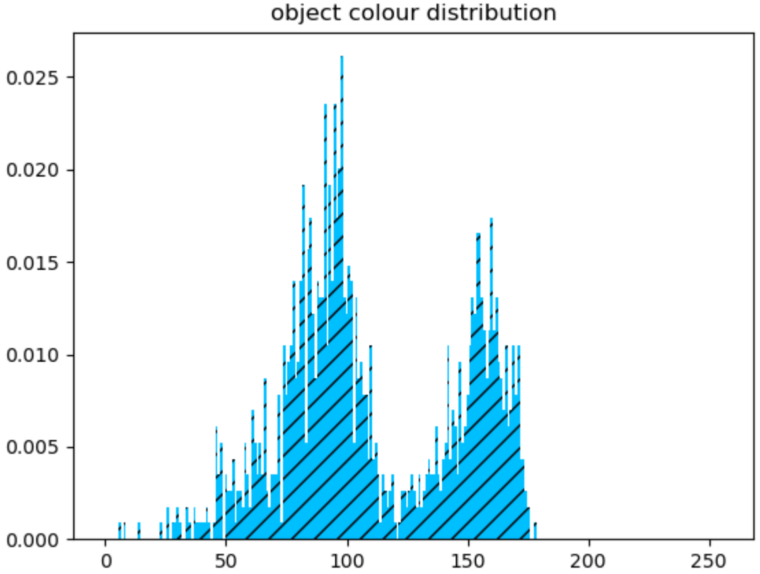
- two kinds of links: t-link (links between pixels and Source or Target) and n-link (links between adjacent pixels)
- thickness of the link indicates its weight - the higher the weight is, the thicker the link is
- t-links are directed, while n-links are undirected
- make an n-link thick if values of the two pixels connected by it are closed
- make a t-link from Source to a pixel thick if the pixel likely belongs to the object
- make a t-link from a pixel to Target thick if the pixel likely belongs to the background
- remove (cut) thin links (including both t-links and n-links) to divide the graph into two independent parts with each has a terminal inside
- the summation of weights of removed links should be minimum, also known as **min-cut problem**

how to assign weights to links?

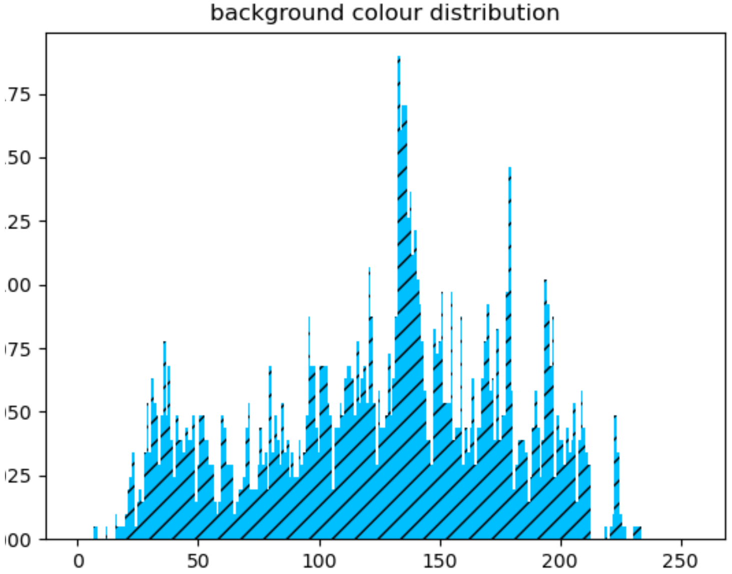
how to solve the min-cut problem?

Preliminary: colour distribution





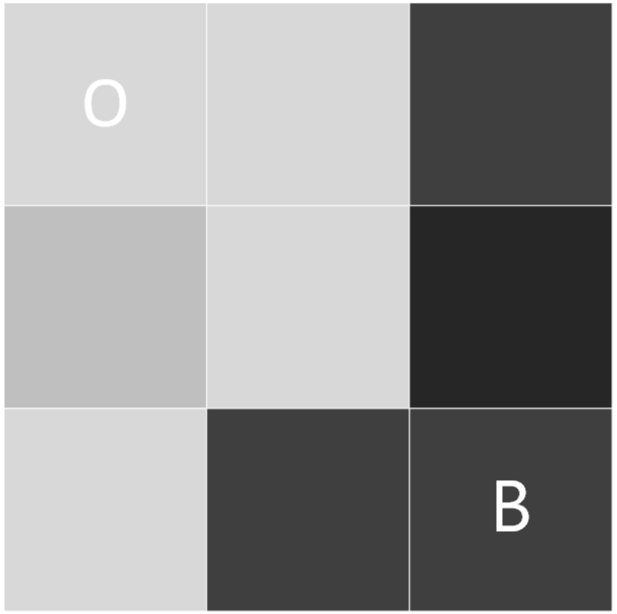
P(I|Object)

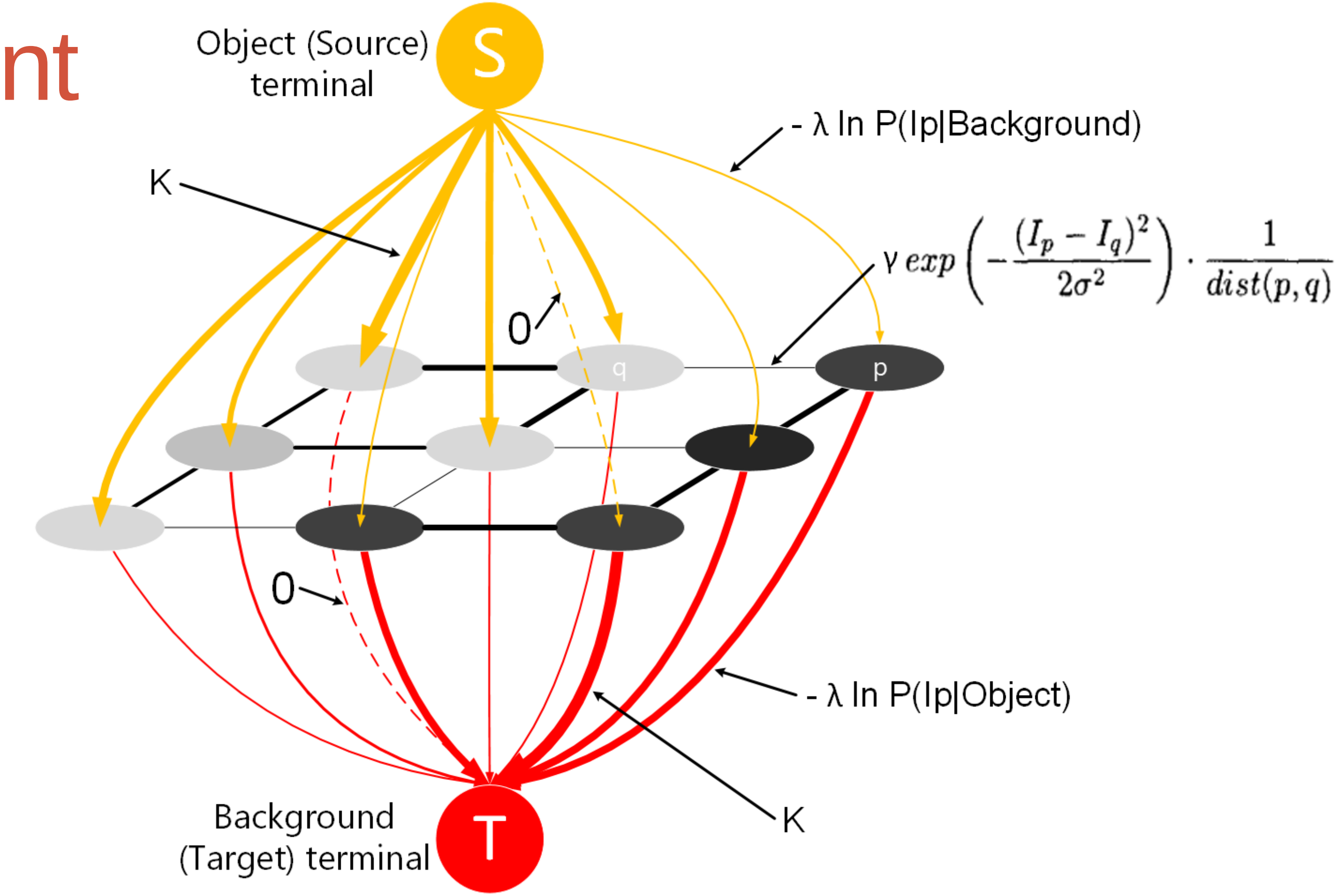


P(I|Background)

calculate colour distributions of pixels covered by red curves (foregound) and blue curves (background) separately

Weight assignment




Object (Source) terminal
S
- λ ln P(Ip|Background)
K
$\gamma\, exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)}$
0
q
p
0
- λ ln P(Ip|Object)
K
Background (Target) terminal
T

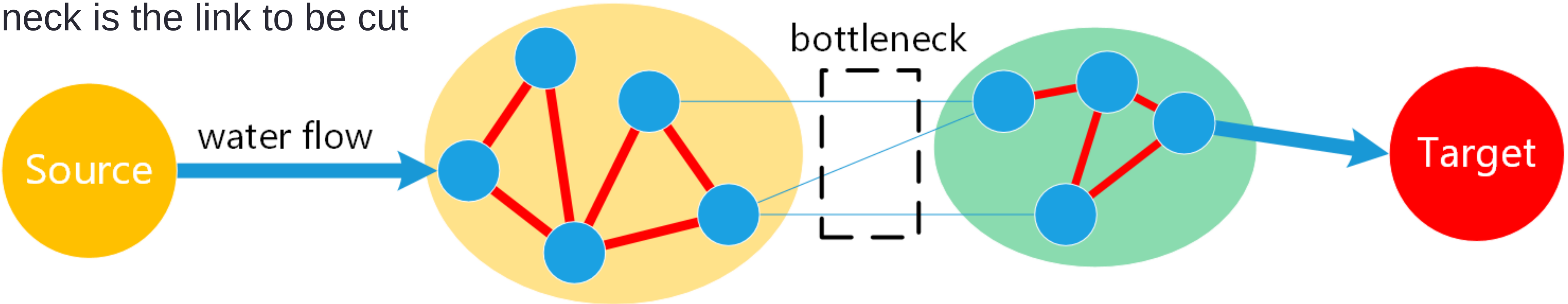
- λ and γ are hyper-parameters

- K should be a large value, can be determined by the formula below

$$K = 1 + \max_{p \in \{\text{pixels in the image}\}} \sum_{q \in \{\text{p's neighbours}\}} \gamma\, exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)}$$

Solving min-cut problem

- min-cut problem is equivalent to max-flow problem
- imagine each link as a tube with different capacities - larger weights, larger capacities
- water is pumped from the Source to the Target, try to maximize the water flow to find the bottleneck
- bottleneck is the link to be cut



An efficient algorithm for solving max-flow problem:

Boykov, Yuri, and Vladimir Kolmogorov. "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision." IEEE transactions on pattern analysis and machine intelligence 26.9 (2004): 1124-1137.

For further details about Graph Cut, read this paper:

Boykov, Yuri Y., and M-P. Jolly. "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images." Proceedings eighth IEEE international conference on computer vision. ICCV 2001. Vol. 1. IEEE, 2001.