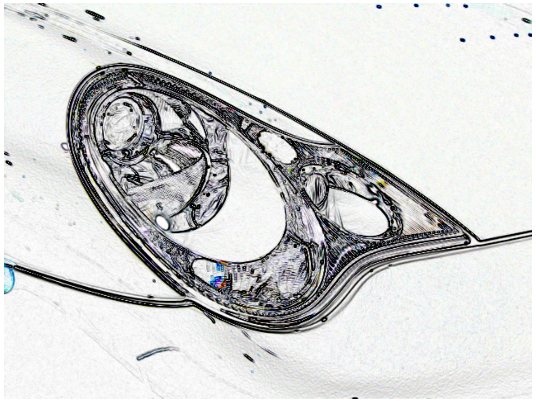
CSE578: Computer Vision

Spring 2017:

Image Segmentation


Anoop M. Namboodiri

Center for Visual Information Technology

IIIT Hyderabad, INDIA

Three "Urges" on seeing a Picture*

1. **To group** proximate and similar parts of the image into meaningful "regions".

   Called segmentation in computer vision.

2. **To connect to memory** to recollect previously seen "objects".

   Called recognition in computer vision.

3. **To measure** quantitative aspects such as number and sizes of objects, distances to/between them, etc.

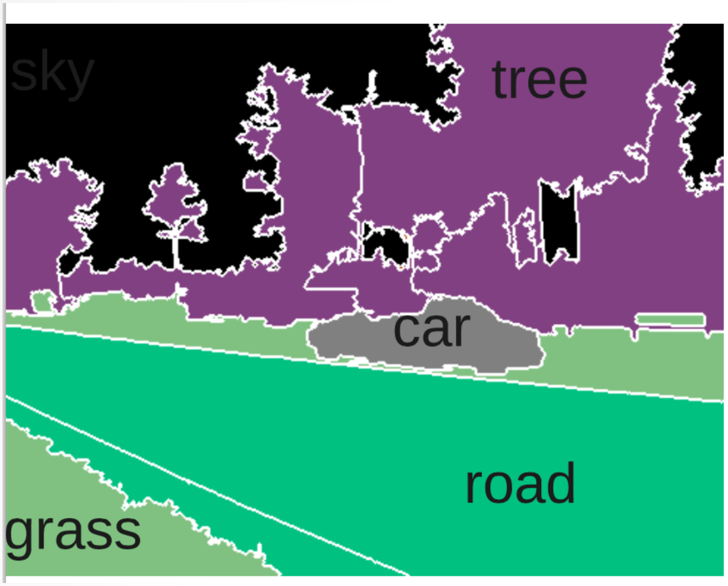
   Called reconstruction in computer vision.

**Jitendra Malik; Mysore Park, Dec. 2011*

Urge to Group

- We don’t see individual pixels (like the computer does!).
- We see groups of pixels together.
- What is the basis for “correct” grouping?

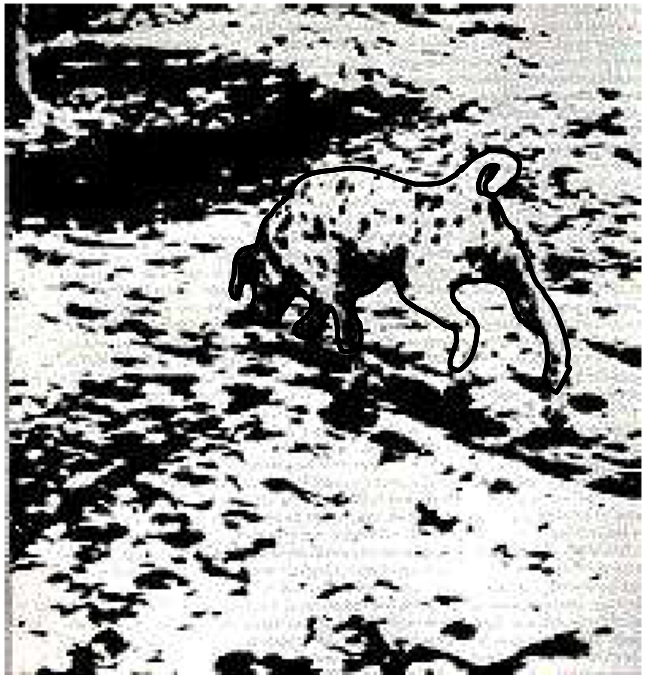
Urge to Group



- Group similar pixels together as objects.
- Group semantically meaningful pixels together as objects.
- Is appearance similarity the same as semantic similarity?

Segmentation

- Dividing an image into semantically meaningful regions.



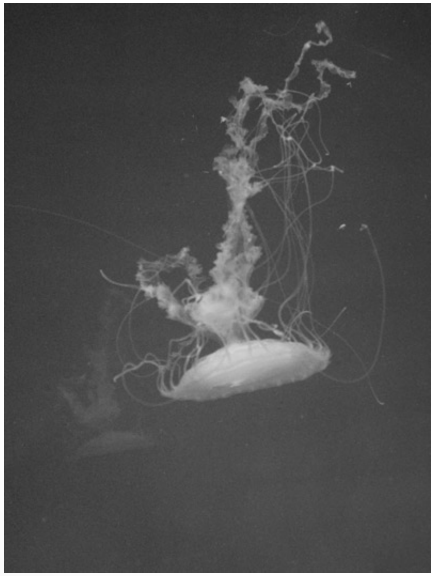
Types of Segmentation

- Classification-based
  - Label pixels based on region properties
  - Label each pixel based on object models
- Region-based
  - Region growing and splitting
- Boundary-based
  - Find edges in the image and use them as region boundary
- Motion-based
  - Group pixels that have consistent motion (e.g., move in the same direction)

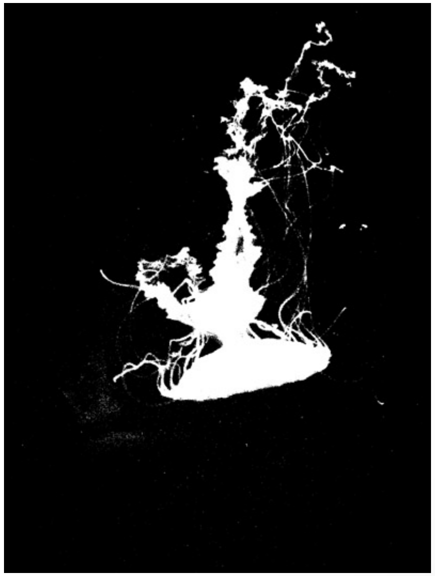
Thresholding

Decide each pixel to be part of an object or background depending on its gray value

$$t(m,n) = \begin{cases} 1 & \text{if } u(m,n) > T \\ 0 & \text{if } u(m,n) \le T \end{cases}$$



Original



Thresholded (T=95)

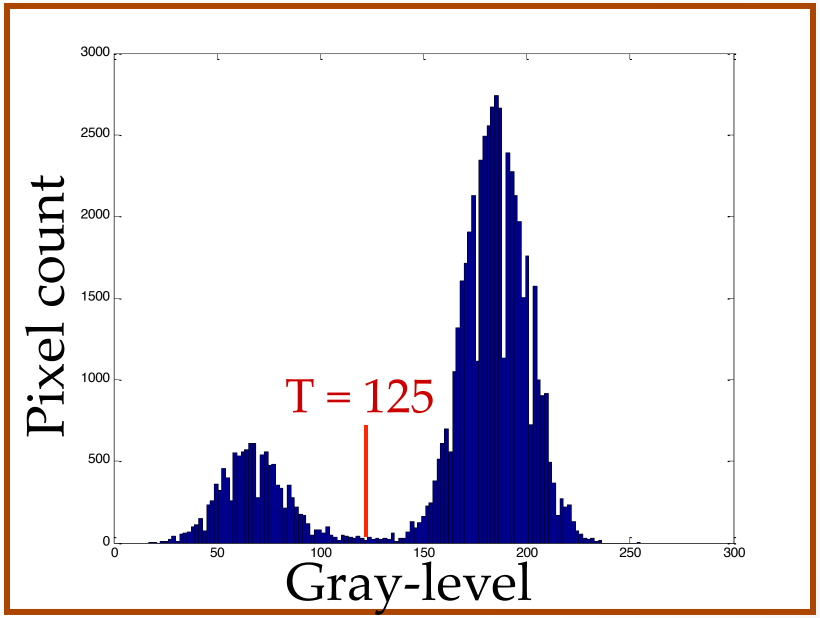
Types of Thresholding

- Global
  - A single threshold is used for the whole image
  - How to determine the threshold?
- Adaptive (Local)
  - Decide the threshold for every pixel depending on its neighborhood
  - How to define the threshold function?

Histogram

- A count of pixels of each graylevel (or range of graylevels) in an image

Grayscale Image



Histogram

Thresholded Image

Original

Thresholded (T=125)

Automatic Thresholding

1. Select an initial estimate of T

2. Segment the image using T. Compute the mean gray values of the two regions, μ_1 and μ_2

3. Set the new threshold $T=(\mu_1+\mu_2)/2$

4. Repeat 2 and 3 until T stabilizes

Assumptions: normal distribution, low noise

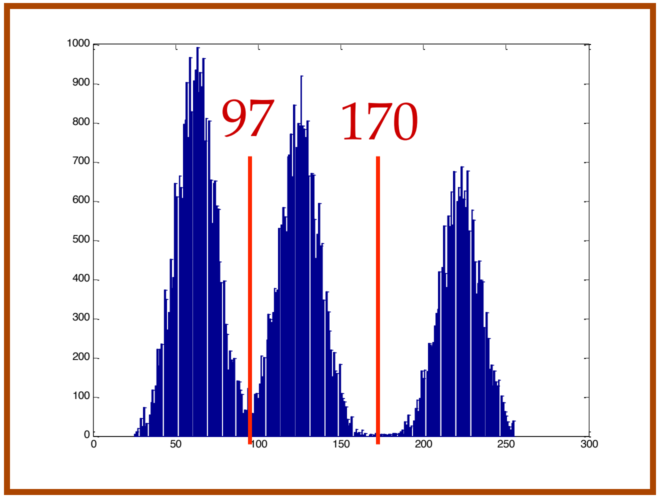
Extensions

- Multiple Thresholds
  - Find multiple peaks and valleys in the gray level histogram
- Multi-spectral Thresholding
  - In color images, one could use different thresholds for each of the color channels

One might set all the background pixels to black, while leave the foreground at the original value so that the information is not lost.

Multiple Thresholds

Original

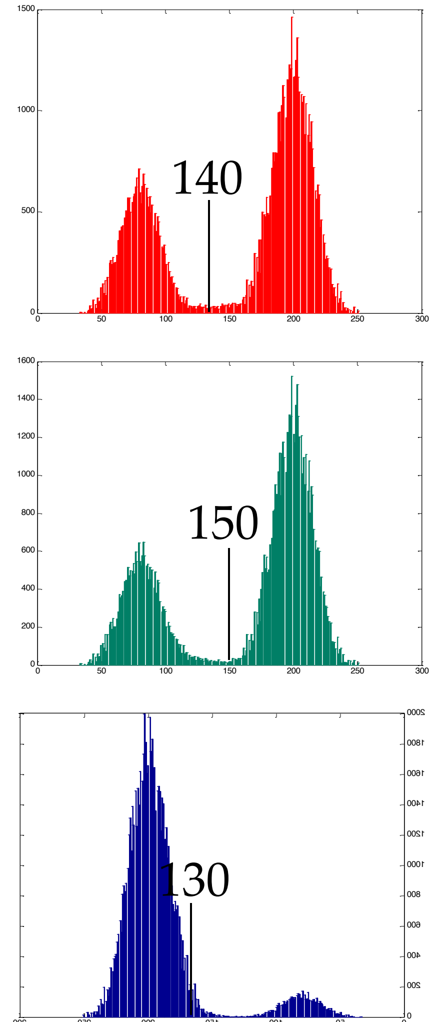


Histogram

Thresholded

Multi-spectral Thresholding

Original



Histograms

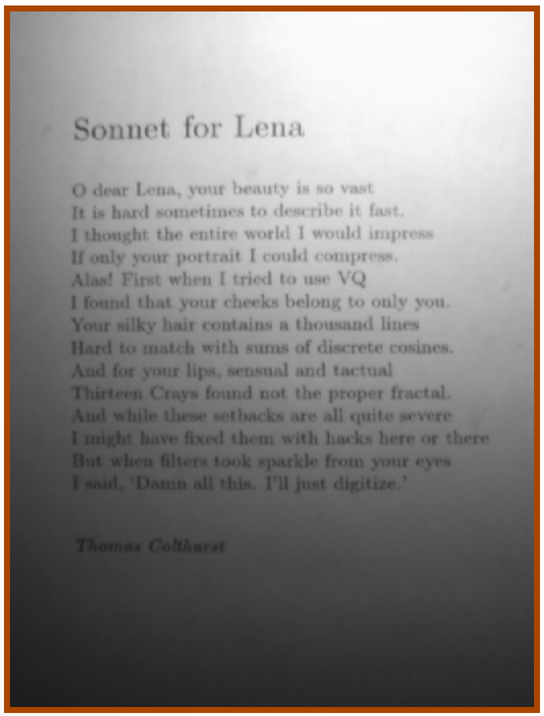
Thresholded

Otsu's Method

1. Compute histogram and probabilities of each intensity
2. Set up initial $\omega_i(0)$ and $\mu_i(0)$ and
3. Step through all possible thresholds $t = 1..t_{max}$
   1. Update ω_i and μ_i
   2. Compute $\sigma^2{}_b(t)$
4. Final threshold corresponds to the maximum $\sigma^2{}_b(t)$
5. Compute two maxima (and respective thresholds t_1 and t_2) using > and >= (first and last maxima)
6. Desired threshold = $(t_1 + t_2)/2$

Adaptive Thresholding

Adaptive thresholding changes the threshold dynamically over the image. This can accommodate strong illumination gradients and shadows

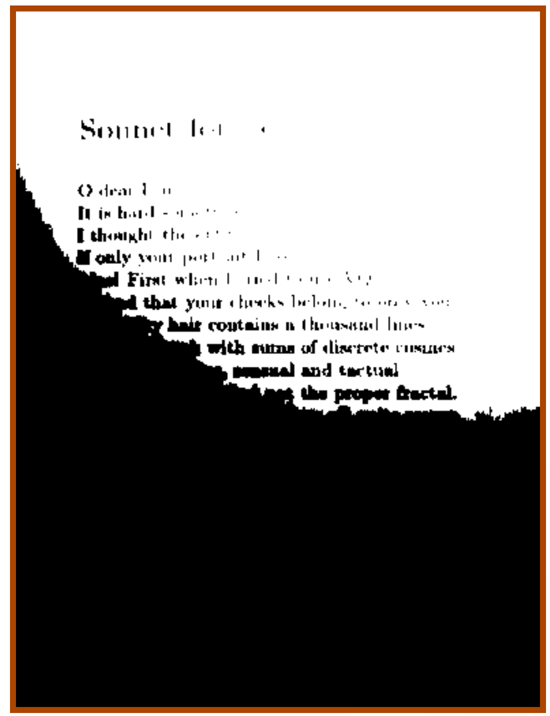


Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

Original

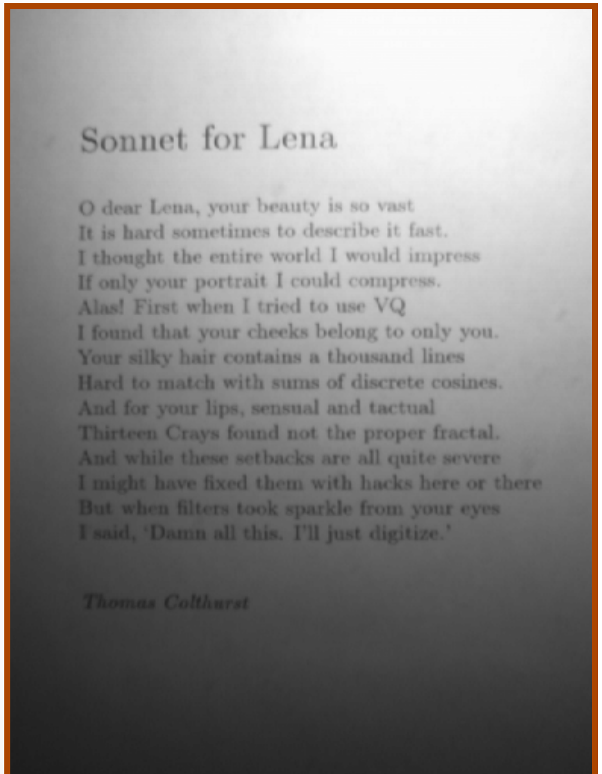


Single Threshold

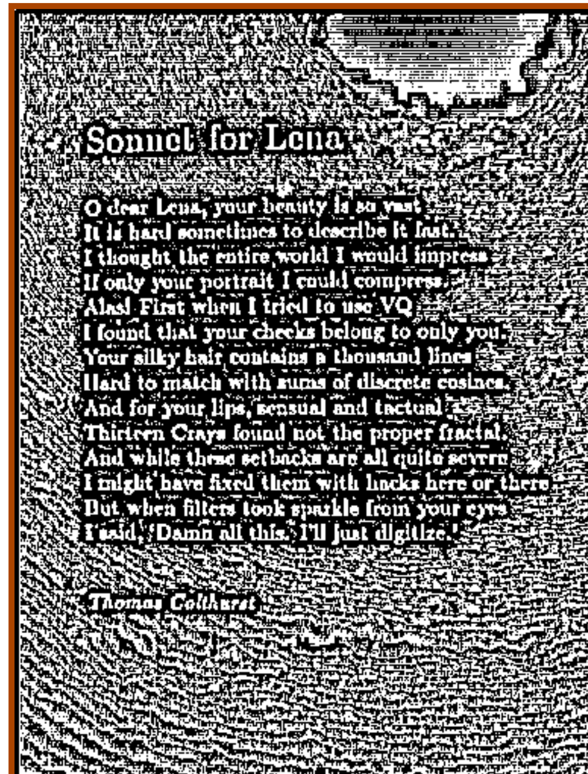
http://www.dai.ed.ac.uk/HIPR2

Adaptive Thresholding

Set the threshold as mean of pixels (gray values) in a neighborhood (say 7x7)



Original



Adaptive Threshold

http://www.dai.ed.ac.uk/HIPR2

Adaptive Thresholding

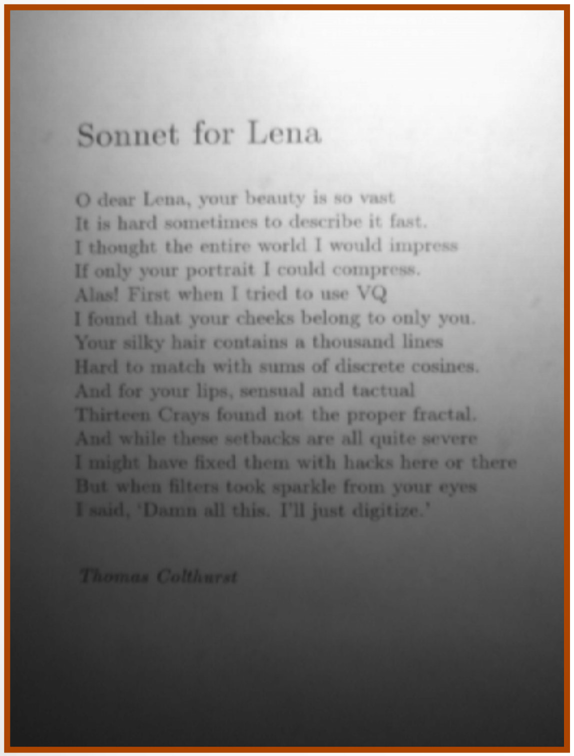
- Thresholding using Mean-C
  Set cxc image regions of uniform graylevel to background
- Chow and Kaneko

1. Apply the mean operator (low pass filter)
2. Subtract original image from the "mean" mage
3. Threshold image in step 2
4. Invert the result

C.K. Chow and T. Kaneko Automatic Boundary Detection of the Left Ventricle from Cineangiograms, Comp. Biomed. Res.(5), 1972, pp. 388-410.

Adaptive Thresholding

<u>Chow & Kaneko Thresholding:</u>



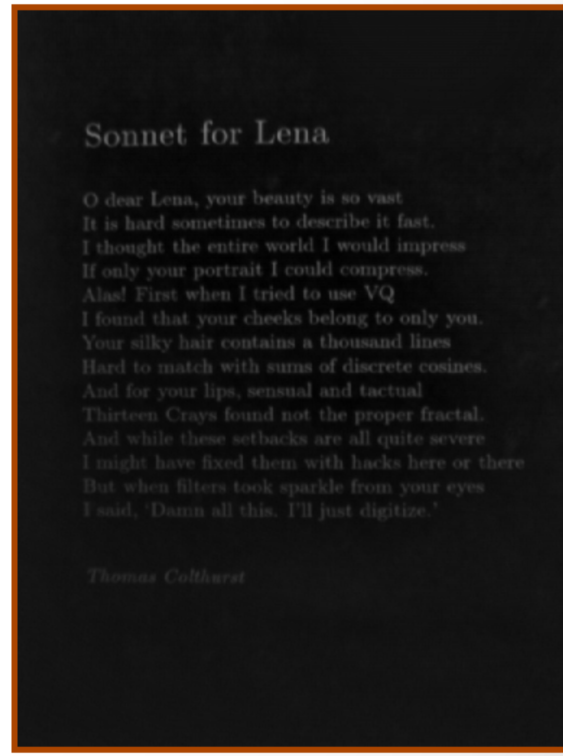
Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

Original

Low-pass filtered



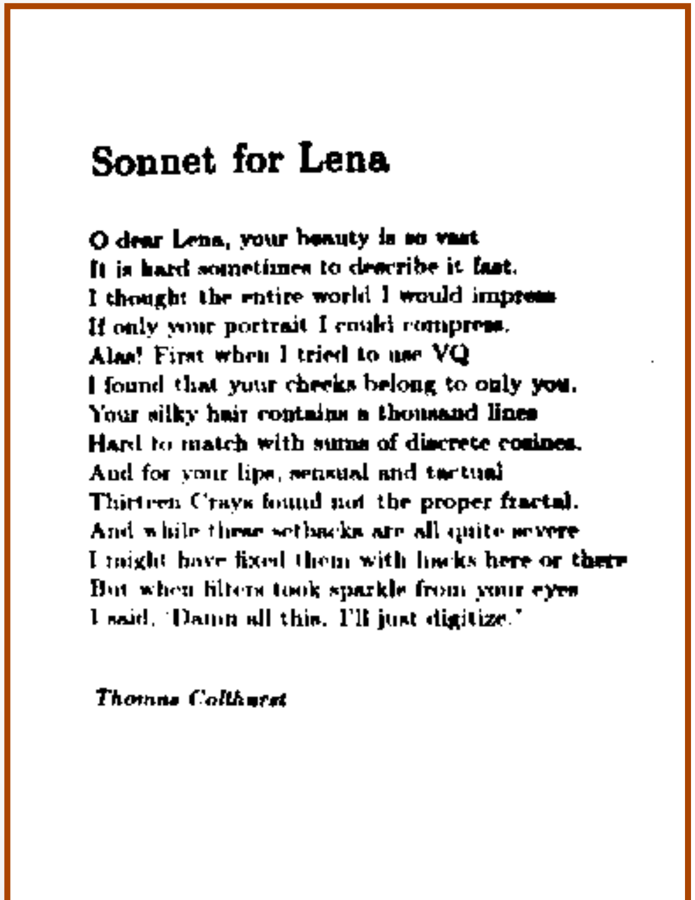
Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

Difference

Adaptive Thresholding Results

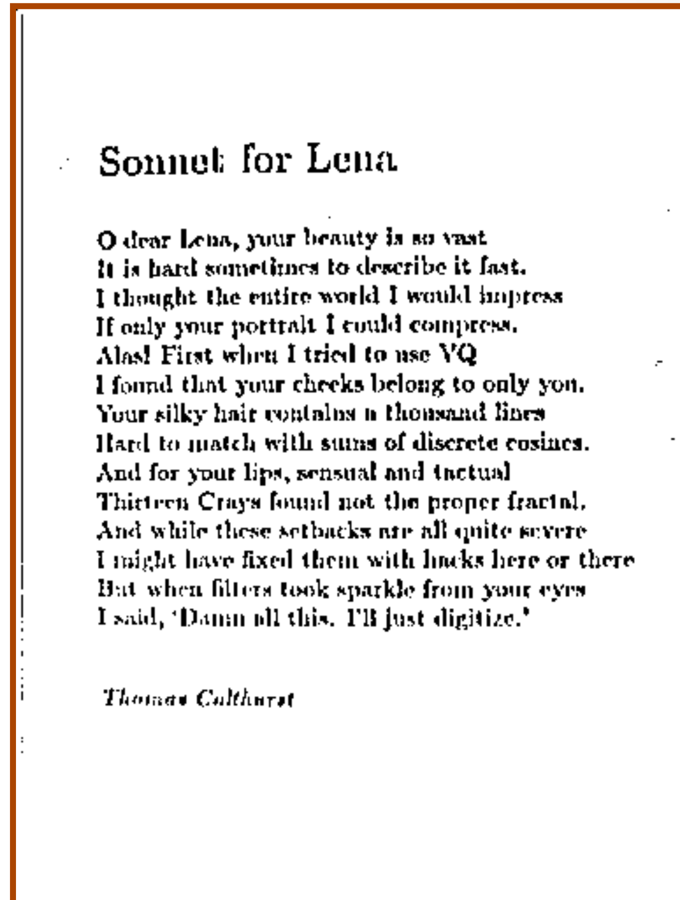


Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

Chow & Kaneko Thresholding



Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

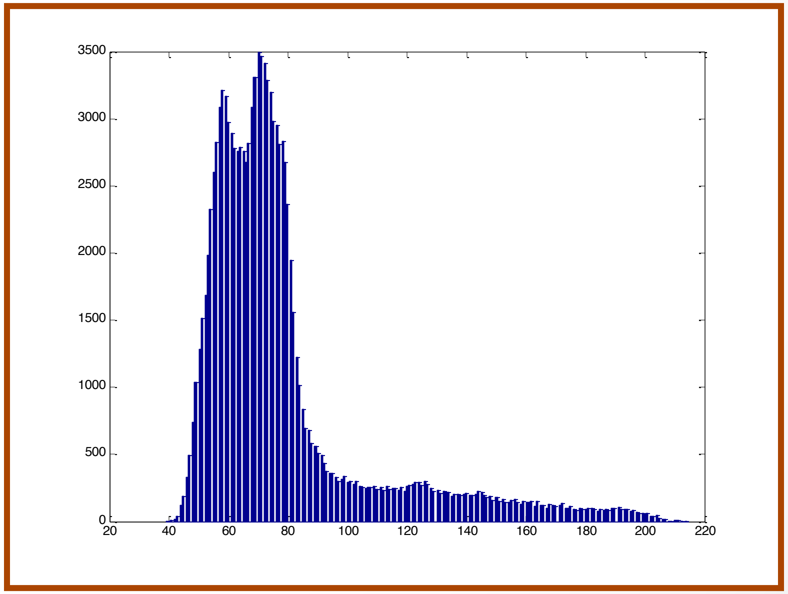
Thomas Colthurst

Mean-C (10) Thresholding

Optimal Thresholding

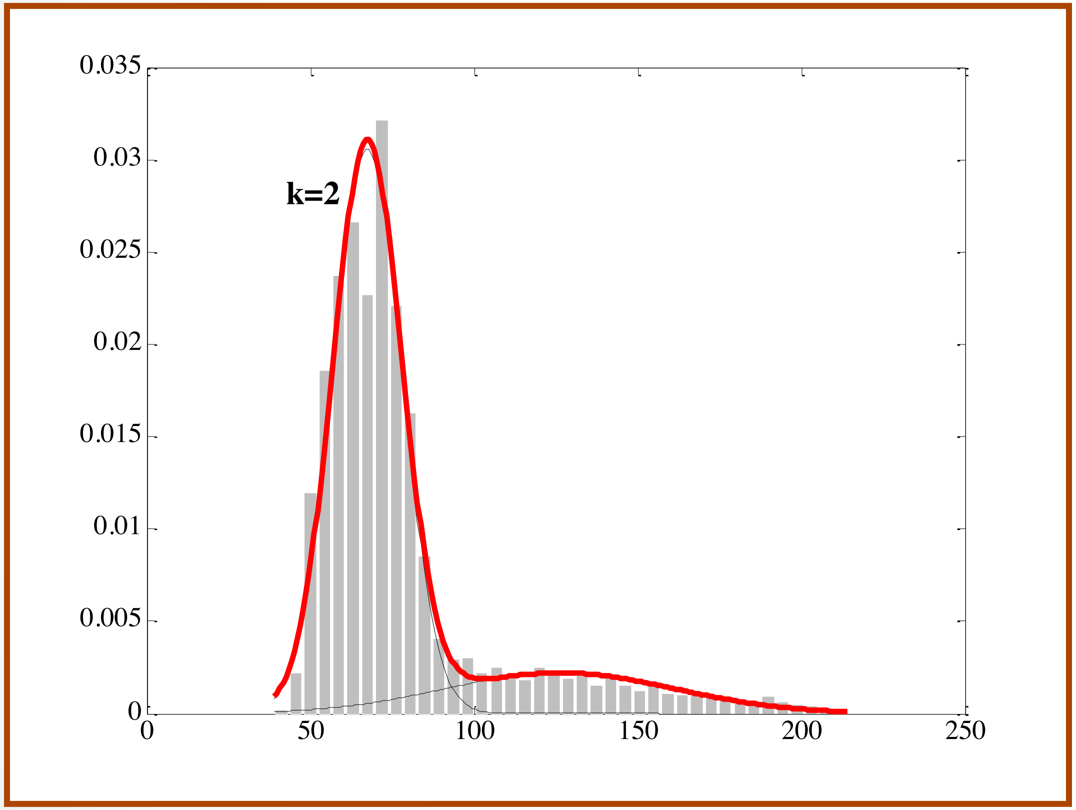
- The graylevel histogram is approximated using a mixture of two gaussian distributions and set the threshold to minimize the segmentation error

Grayscale Image



Histogram

Optimal Thresholding



Histogram with bimodal fit

Thresholded (T=94)

Gaussian Mixture Estimation by EM

- Obj: = $N(\mu_1,\sigma_1) = \frac{1}{\sigma_1\sqrt{2\pi}} e^{\frac{(x-\mu_1)^2}{2\sigma_1^2}}$
- Bkg: = $N(\mu_2,\sigma_2) = \frac{1}{\sigma_2\sqrt{2\pi}} e^{\frac{(x-\mu_2)^2}{2\sigma_2^2}}$
- E-Step: Computed the expected pixel label assignments.
- M-Step: Computed Maximum-Likelihood estimates of the parameters: $\mu_1,\sigma_1,\mu_2,\sigma_2$

Segmentation as Optimal Labeling

- Model knowledge about the world
- Classify each pixel as belonging to a specific object
    - Independent classification does not work
    - Need to incorporate neighborhood information
- Consider a graph over the image
    - Each node in the graph need to be labeled
    - Edges in the graph represent neighborhood constraints
- Define a cost function, $Q(f)$, using the above
- Compute the optimal labeling wrt $Q(f)$.

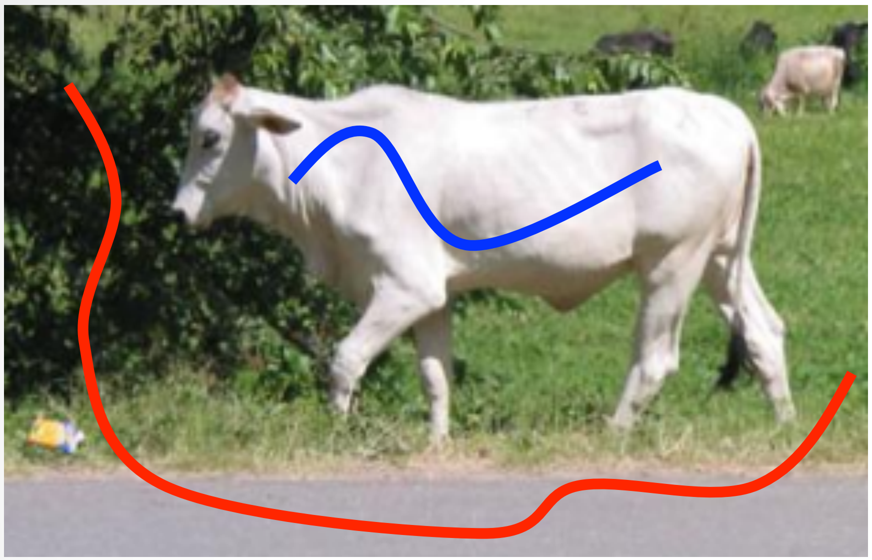
Binary Image Segmentation

How ?

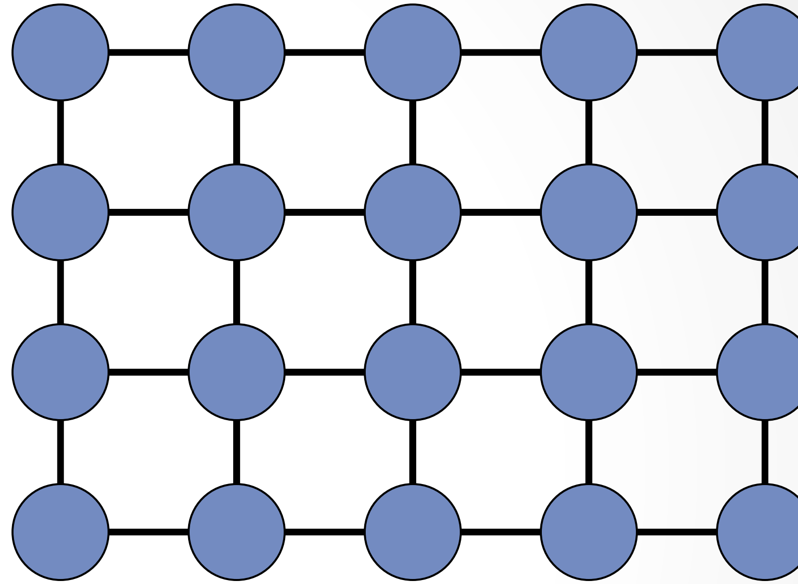
Cost function Models *our* knowledge about natural images

Optimize cost function to obtain the segmentation

Binary Image Segmentation



Object - white, Background - green/grey



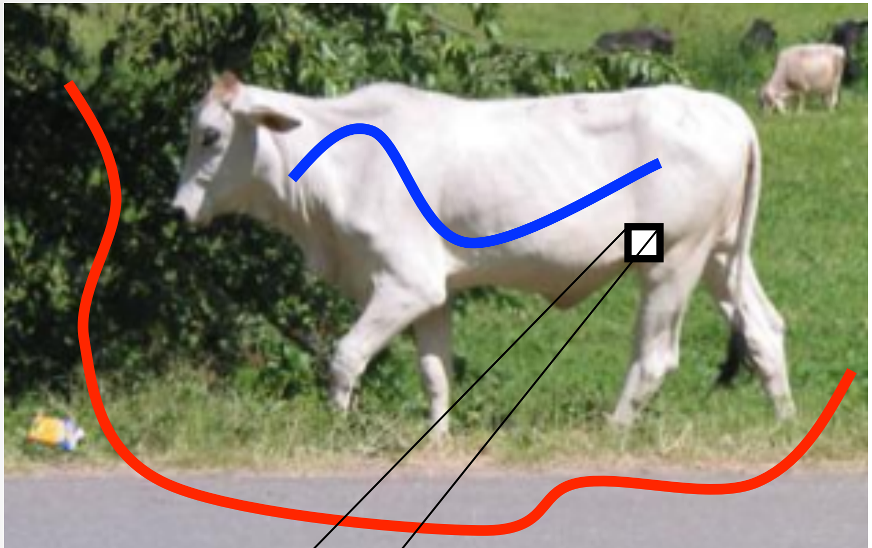
Graph G = (V,E)

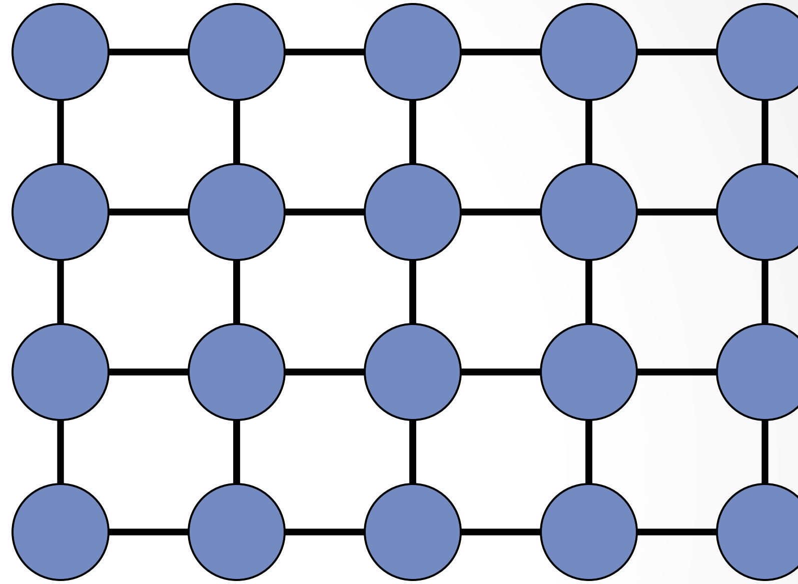
Each vertex corresponds to a pixel

Edges define a 4-neighbourhood *grid* graph

Assign a label to each vertex from L = {obj,bkg}

Binary Image Segmentation





Object - white, Background - green/grey

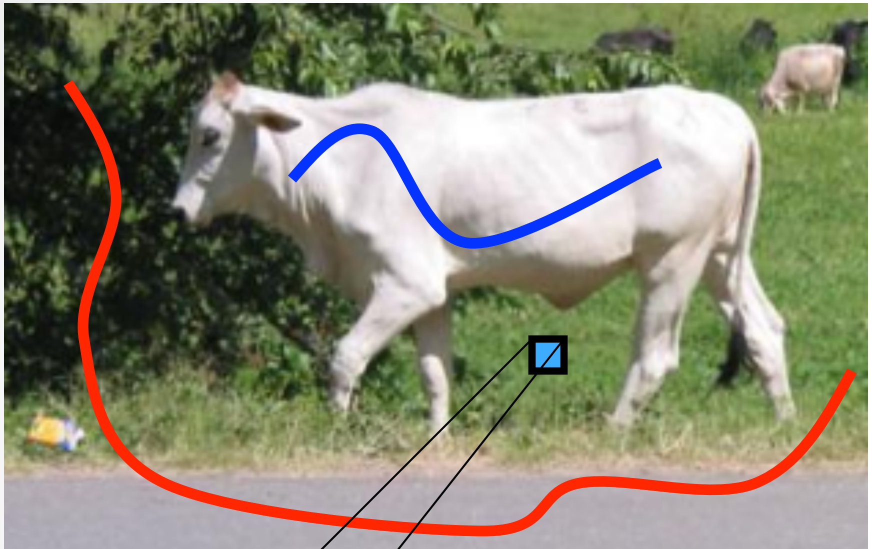
Graph G = (V,E)

Cost of a labelling f : V ➔ L

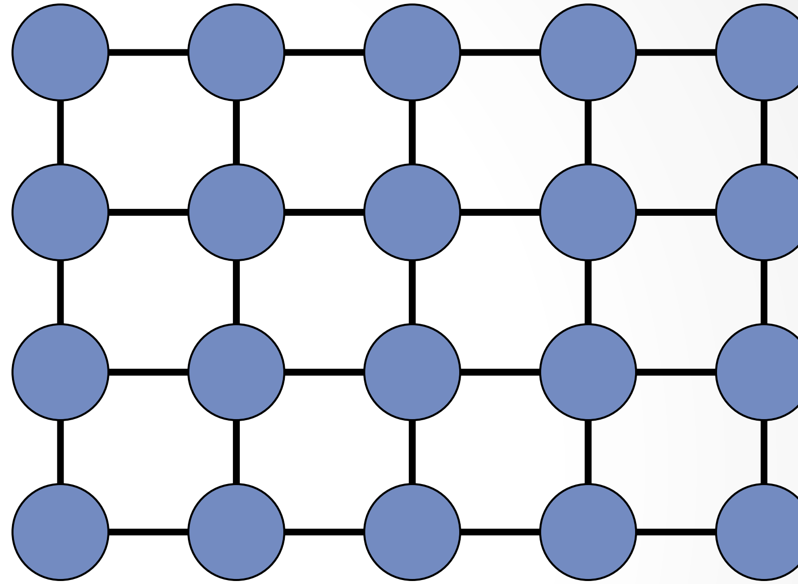
Per Vertex Cost

Cost of label 'obj' low Cost of label 'bkg' high

Binary Image Segmentation



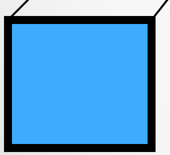
Object - white, Background - green/grey



Graph G = (V,E)

Cost of a labelling f : V ➔ L

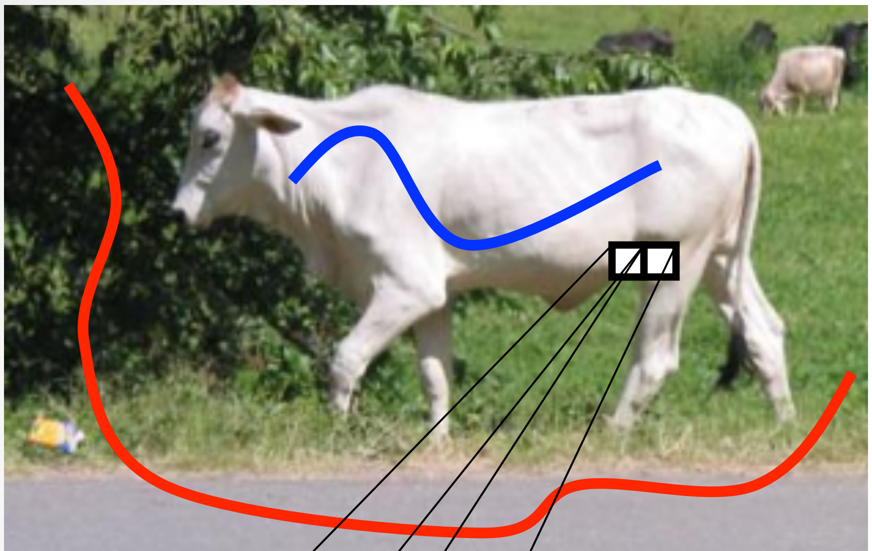
Per Vertex Cost

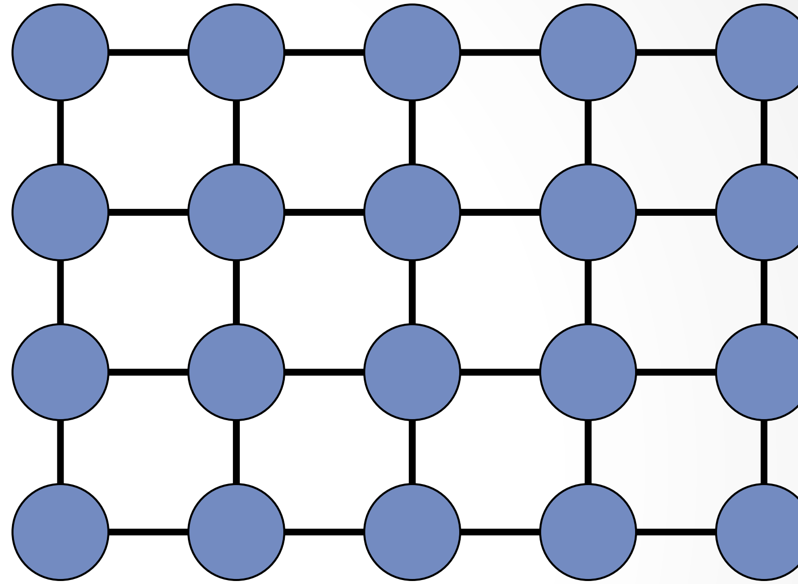


Cost of label ‘obj’ high Cost of label ‘bkg’ low

UNARY COST

Binary Image Segmentation





Object - white, Background - green/grey

Graph G = (V,E)

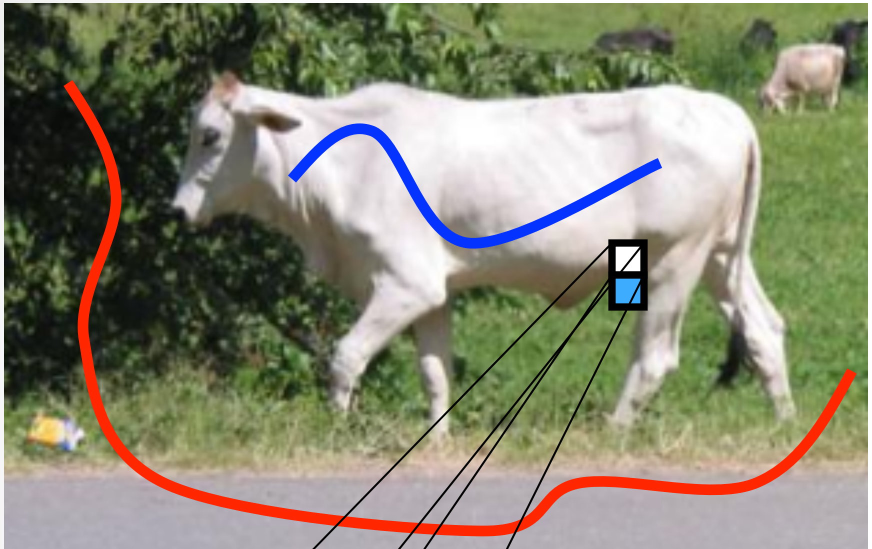
Cost of a labelling f : V ➔ L

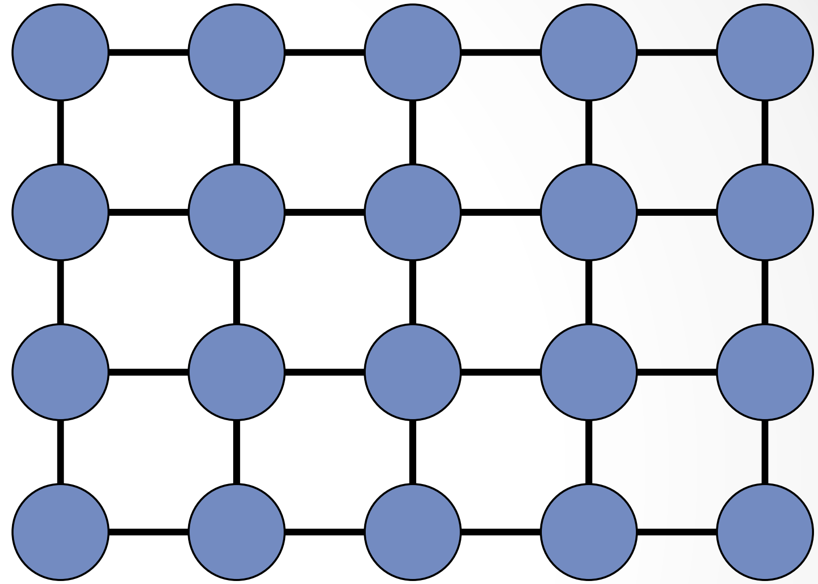
Per Edge Cost

Cost of same label low

Cost of different labels high

Binary Image Segmentation



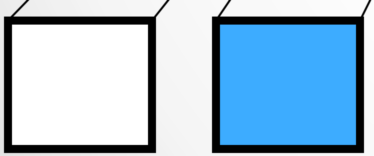


Object - white, Background - green/grey

Graph G = (V,E)

Cost of a labelling f : V ➔ L

Per Edge Cost

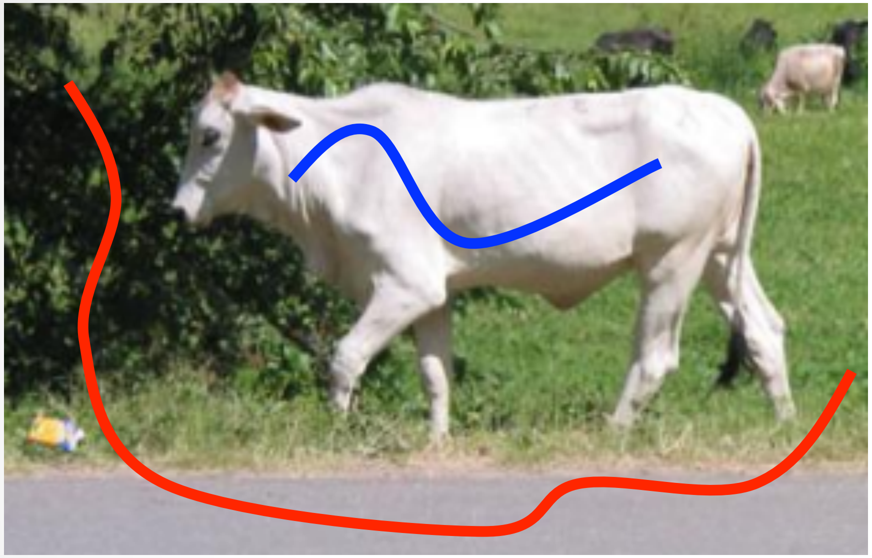


Cost of same label high

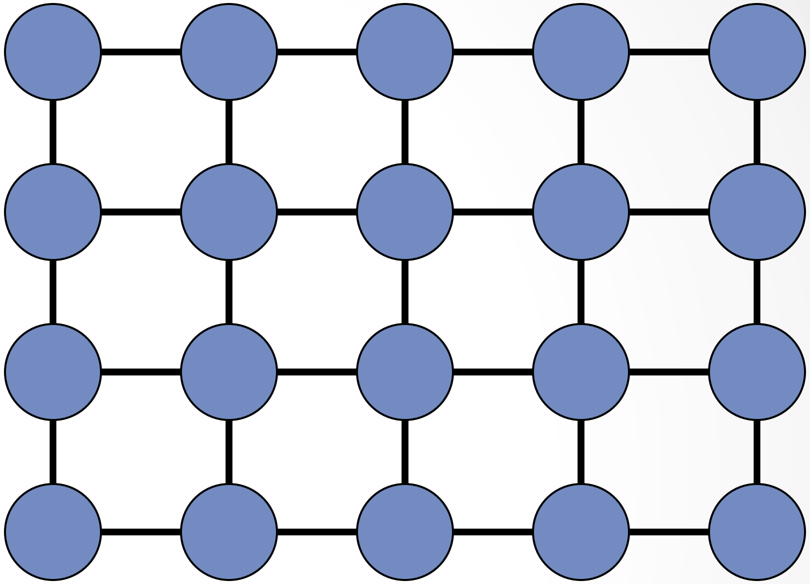
Cost of different labels low

PAIRWISE COST

Binary Image Segmentation



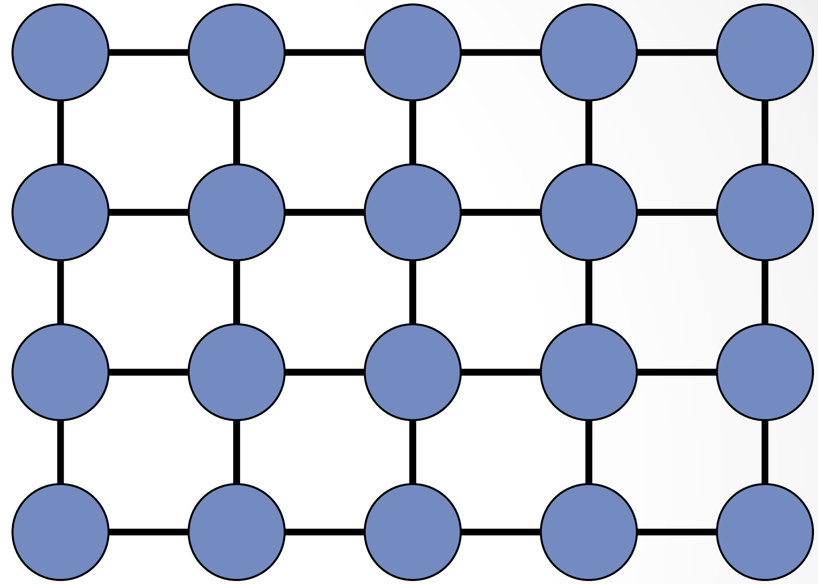
Object - white, Background - green/grey



Graph G = (V,E)

Problem: Find the labeling with minimum cost f*

Binary Image Segmentation



Graph G = (V,E)

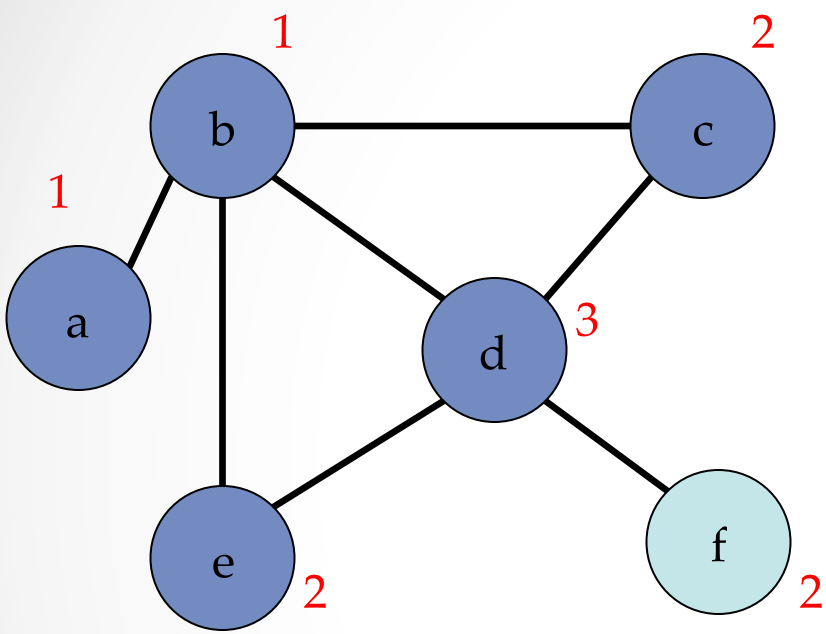
$L = \{fg, bg\}$

Vertex corresponds to a pixel

Edges define grid graph

Problem: Find the labeling with minimum cost f*

The General Problem



Graph G = (V, E)

Discrete label set L = {1,2,…,h}

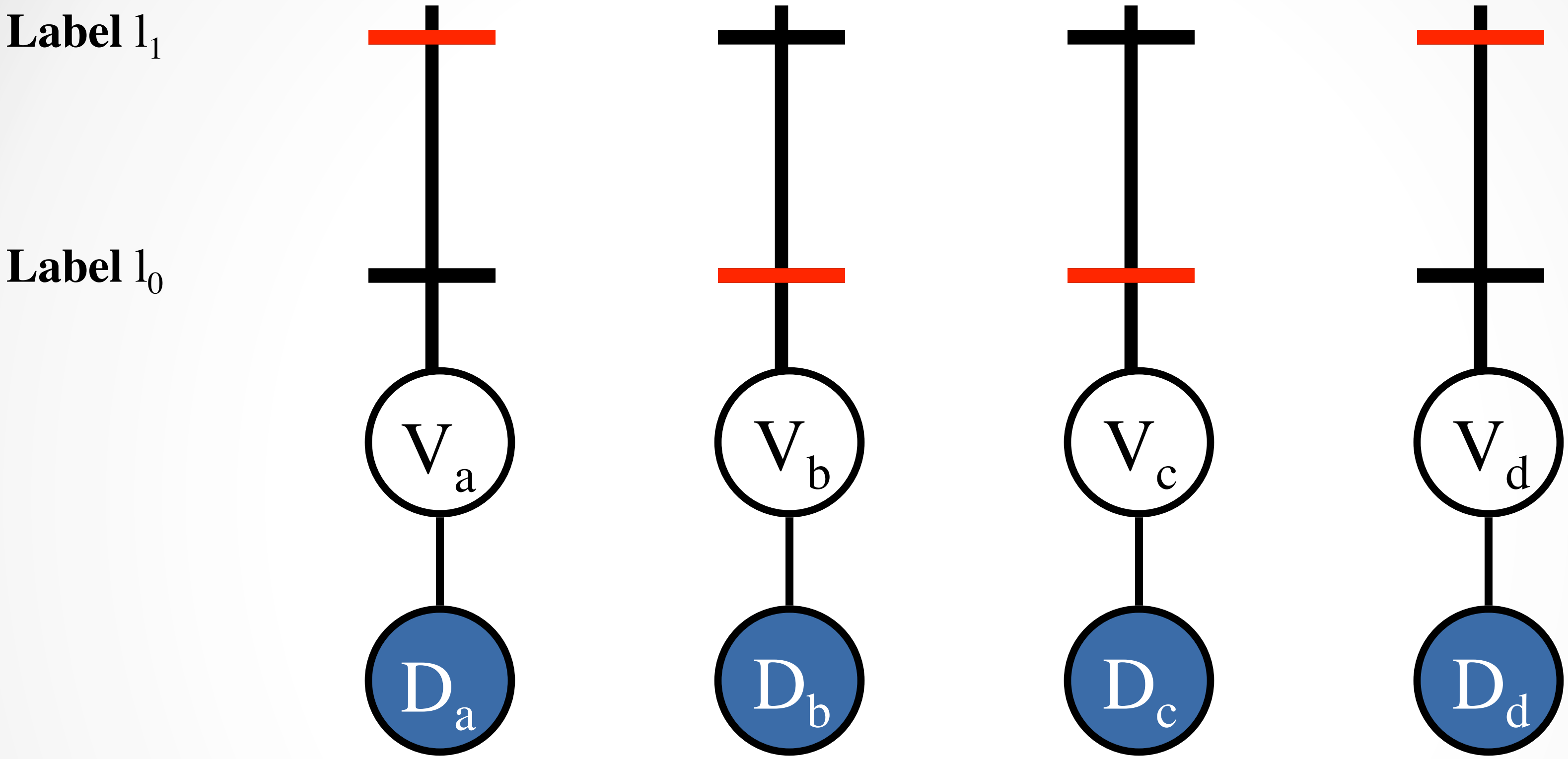
Assign a label to each vertex
$f: V \rightarrow L$

Cost of a labelling Q(f)

Unary Cost Pairwise Cost

Find $f^* = \arg\min Q(f)$

Formulation: Energy Function

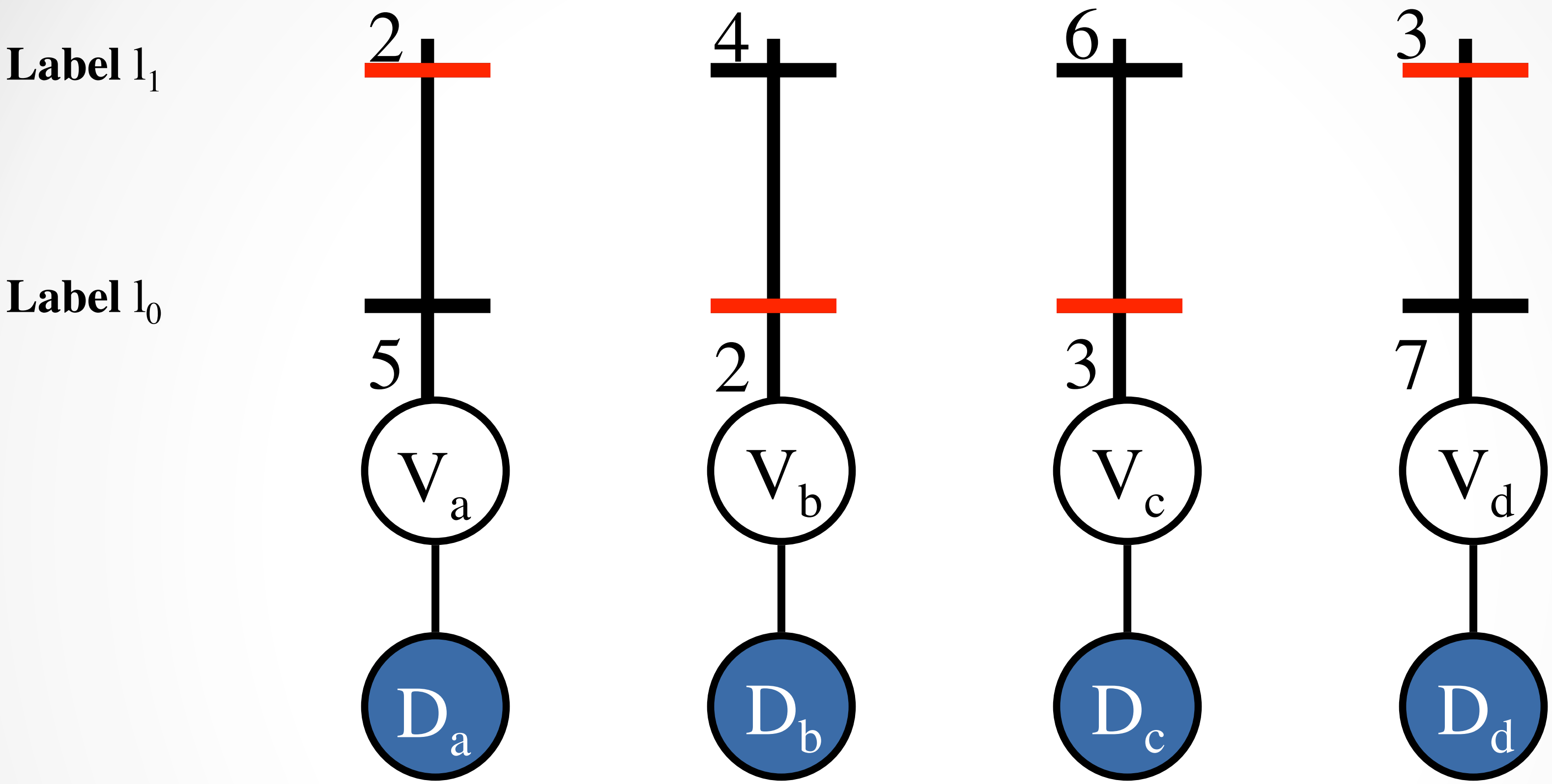


Random Variables $V = \{V_a, V_b, \ldots.\}$

Labels $L = \{l_0, l_1, \ldots.\}$ Data D

Labelling f: $\{a, b, \ldots. \} \rightarrow \{0,1, \ldots\}$

Energy Function



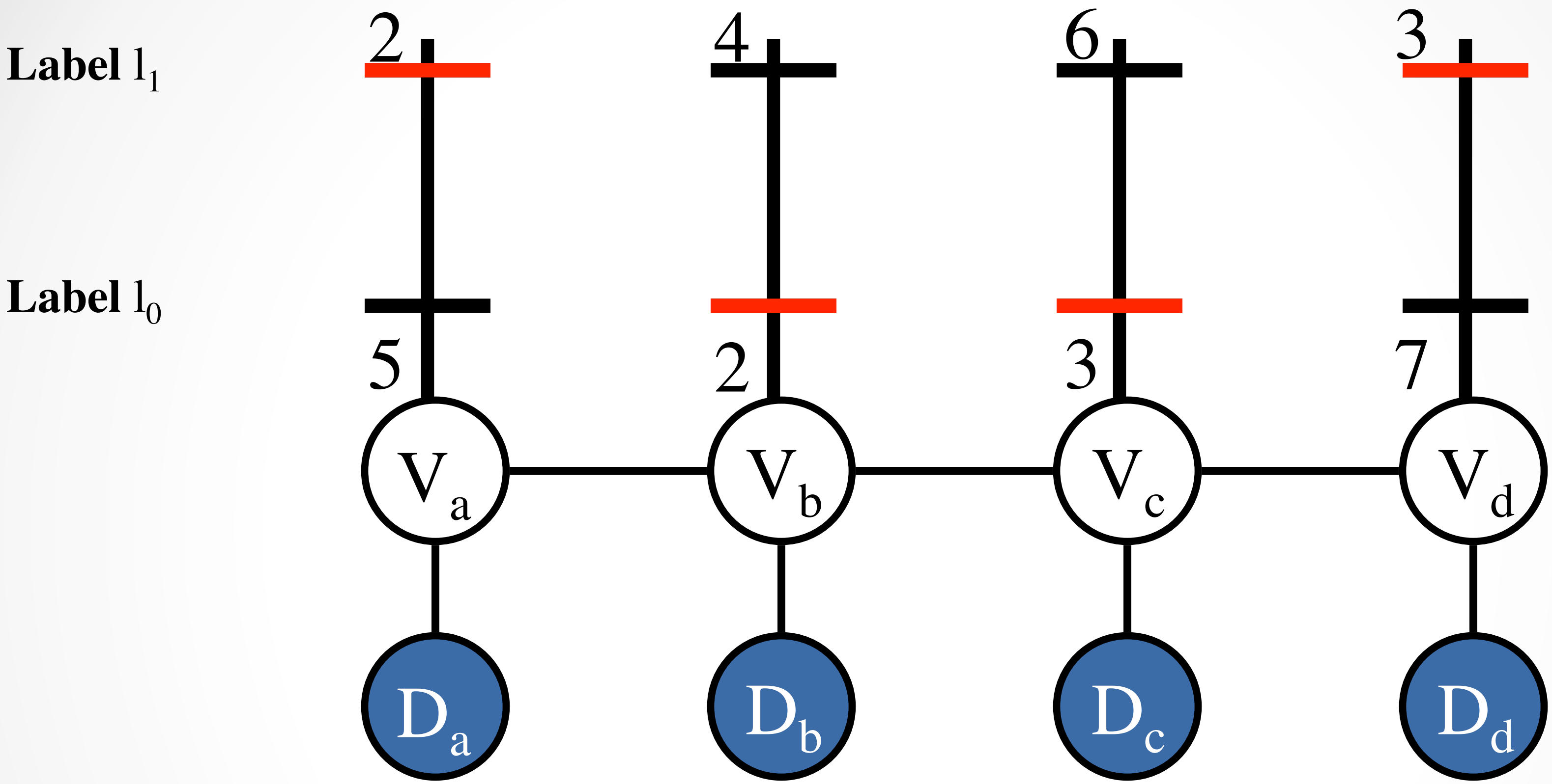
$$Q(f) = \sum_a \theta_{a;f(a)}$$

Easy to minimize

Unary Potential

Neighbourhood

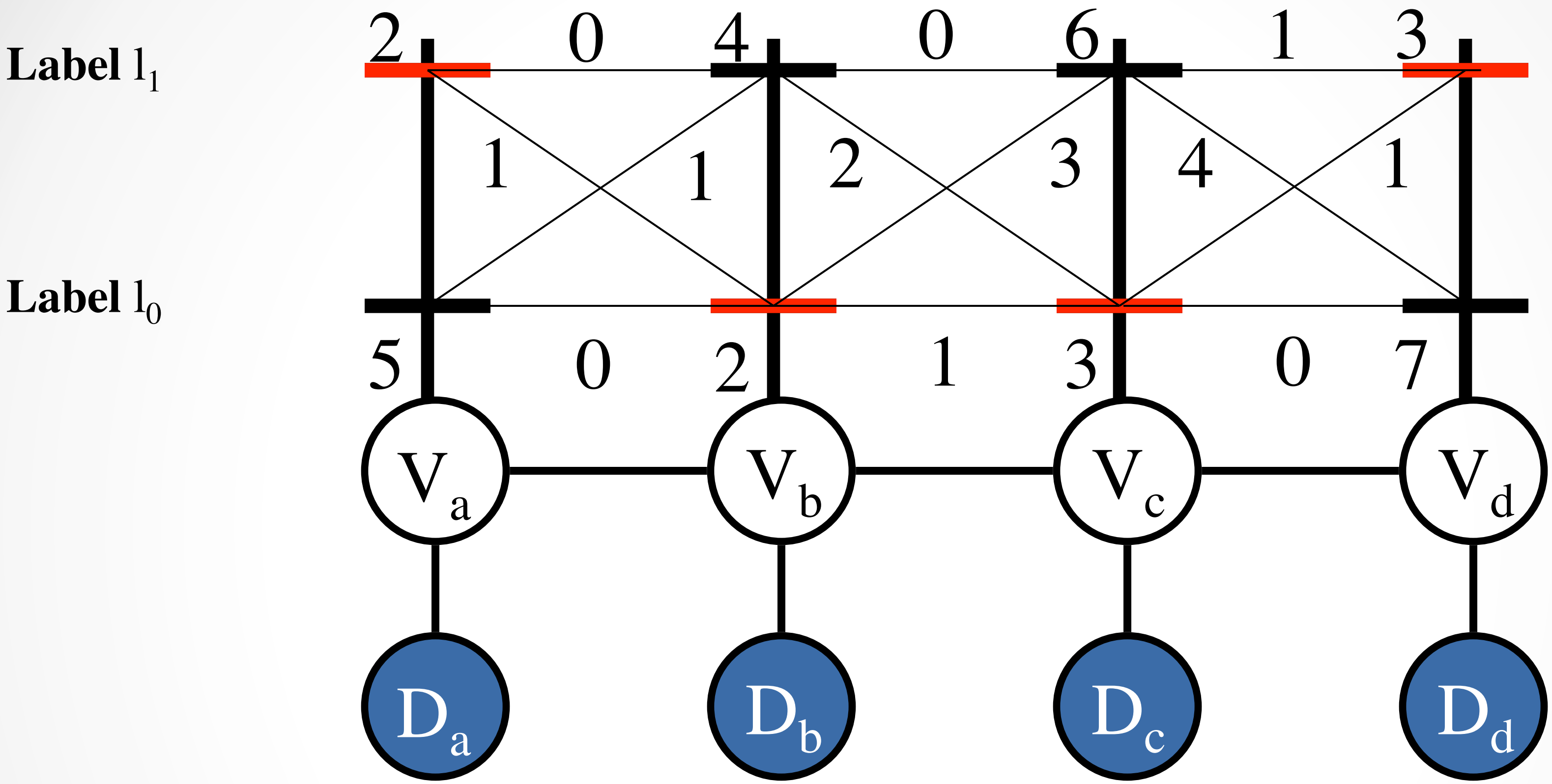
Energy Function



E : (a,b) $\in$ E iff V_a and V_b are neighbours

E = { (a,b) , (b,c) , (c,d) }

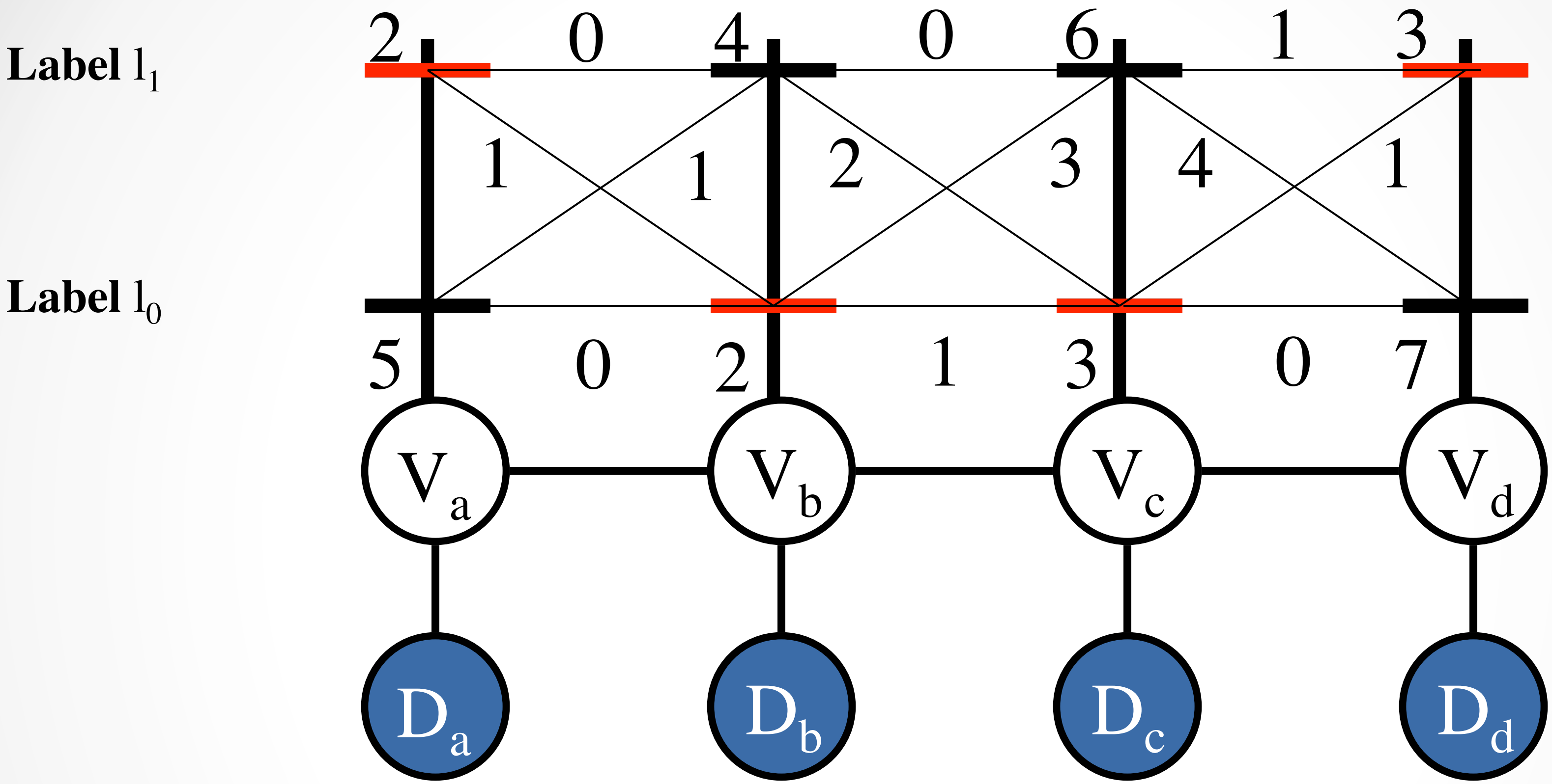
Energy Function



Pairwise Potential

$$Q(f) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$$

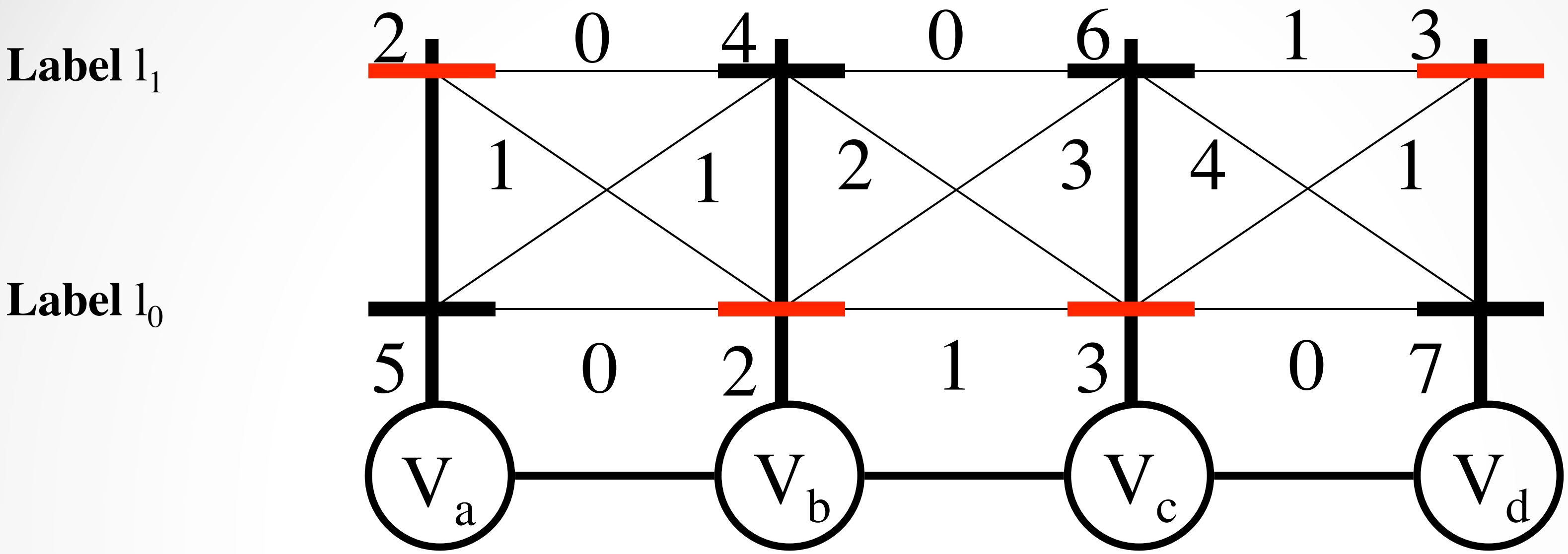
Energy Function



$$Q(f; \theta) = \sum_a \theta_{a;f(a)} \quad + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$$

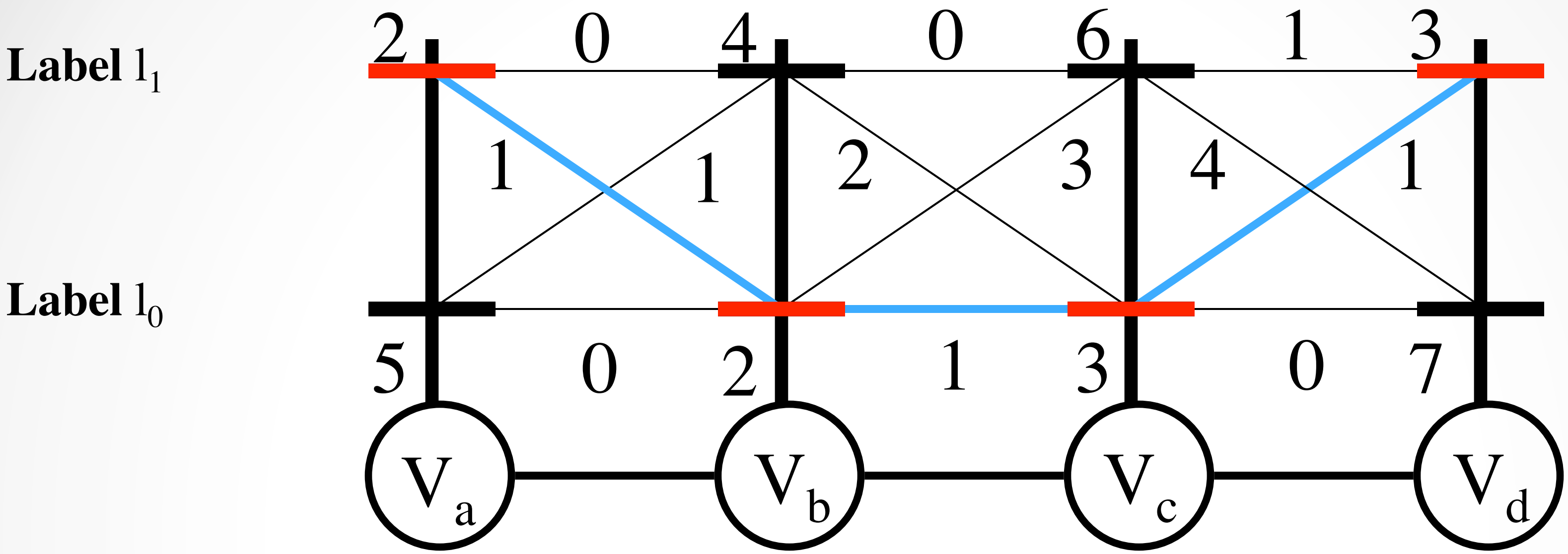
Parameter

MAP Estimation



$$Q(f;\, \boldsymbol{\theta}) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$$

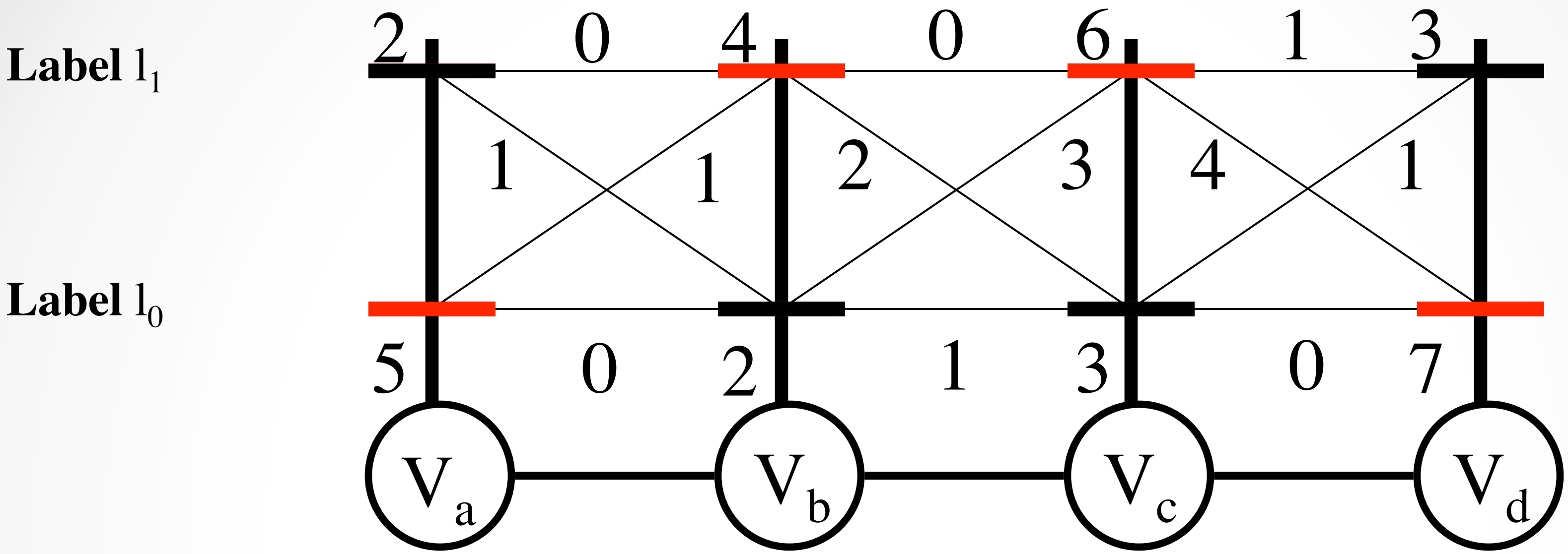
MAP Estimation



$$Q(f; \boldsymbol{\theta}) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$$

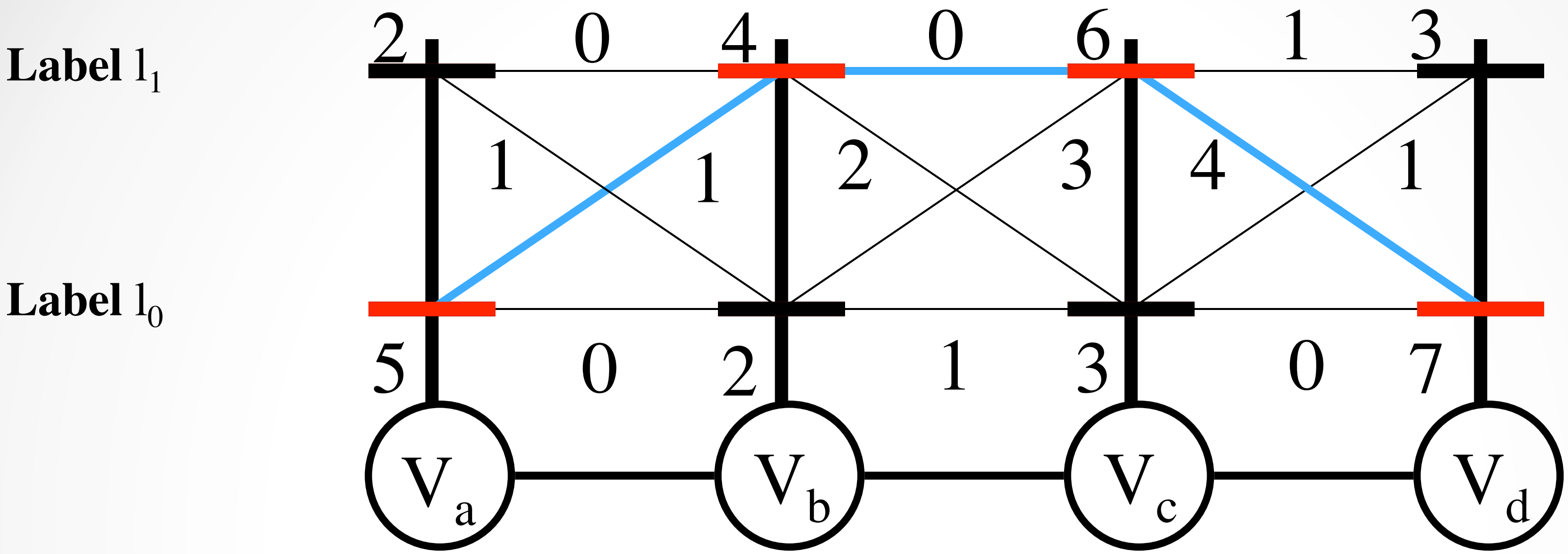
2 + 1 + 2 + 1 + 3 + 1 + 3 = 13

MAP Estimation



$$Q(f; \boldsymbol{\theta}) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$$

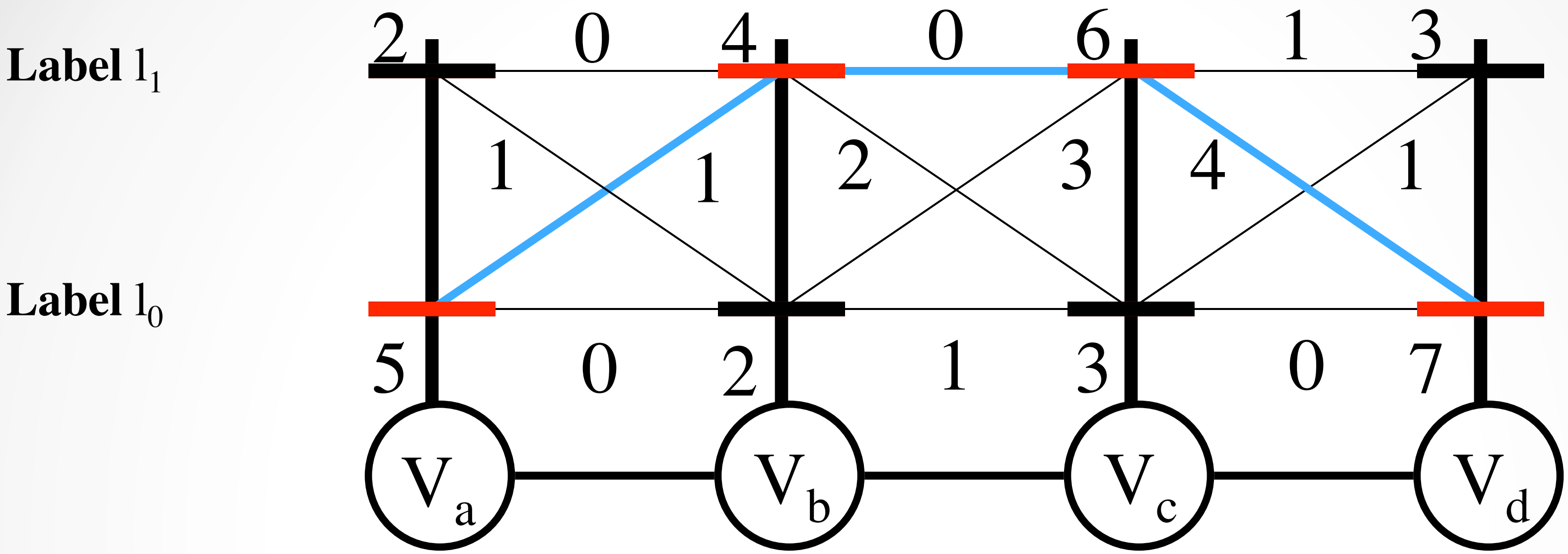
MAP Estimation



$$Q(f;\boldsymbol{\theta}) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$$

5 + 1 + 4 + 0 + 6 + 4 + 7 = 27

MAP Estimation



$$q^* = \min Q(f; \theta) = Q(f^*; \theta)$$

$$Q(f; \boldsymbol{\theta}) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$$

$$f^* = \arg\min Q(f; \theta)$$

MAP Estimation

16 possible labellings

$f^* = \{1, 0, 0, 1\}$

$q^* = 13$

f(a)	f(b)	f(c)	f(d)	Q(f; θ)
0	0	0	0	18
0	0	0	1	15
0	0	1	0	27
0	0	1	1	20
0	1	0	0	22
0	1	0	1	19
0	1	1	0	27
0	1	1	1	20

f(a)	f(b)	f(c)	f(d)	Q(f; θ)
1	0	0	0	16
1	0	0	1	13
1	0	1	0	25
1	0	1	1	18
1	1	0	0	18
1	1	0	1	15
1	1	1	0	23
1	1	1	1	16

Computational Complexity

Segmentation

$2^{|V|}$

$|V|$ = number of pixels ≈ 320 * 480 = 153600

Can we do better than brute-force?

MAP Estimation is NP-hard !!

Computational Complexity

Segmentation

$2^{|V|}$

$|V|$ = number of pixels ≈ 320 * 480 = 153600

Exact algorithms do exist for special cases

Good approximate algorithms for general case