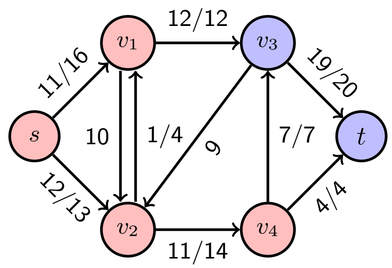
Probabilistic Graphical Models in Computer Vision (IN2329)

Csaba Domokos

Summer Semester 2015/2016

Agenda for today's lecture *

In the **previous lecture** we learnt about the minimum $s - t$ cut problem



Today we are going to learn about

- Exact solution for *binary image segmentation* via *graph cut*
- Approximate solutions for the multi-label problem:
  - α-expansion
  - $\alpha - \beta$ swap

Regular functions $*$

Let us consider a function f of two binary variables, then f is called **regular**, if it satisfies the following inequality

$$f(0,0) + f(1,1) \leqslant f(0,1) + f(1,0) \ .$$

Example: the **Potts-model** is *regular*, i.e.

$$[\![0 \neq 0]\!] + [\![1 \neq 1]\!] = 0 \leqslant 2 = [\![0 \neq 1]\!] + [\![1 \neq 0]\!] \ .$$

Regular energy functions

Let us consider an *energy function* E of n binary variables which can be written as the sum of functions of up to two variables, that is

$$E(y_1, \dots, y_n) = \sum_i E_i(y_i) + \sum_{i<j} E_{ij}(y_i, y_j) \; .$$

E is *regular*, if each term E_{ij} for all $i < j$ satisfies

$$E_{ij}(0,0) + E_{ij}(1,1) \leqslant E_{ij}(0,1) + E_{ij}(1,0) \; .$$

If each term E_{ij} is *regular*, then it is possible to find the **global** minimum of E in *polynomial time* by solving a *minimum $s-t$ cut problem*.

Binary image segmentation

We have already seen that **binary image segmentation** can be reformulated as the minimization of an *energy function* $E : \{0,1\}^{\mathcal{V}} \times \mathcal{X} \to \mathbb{R}$:

$$E(\mathbf{y}; \mathbf{x}) = \sum_{i \in \mathcal{V}} E_i(y_i; x_i) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(y_i, y_j; x_i, x_j) \; .$$

where $\mathcal{V}$ corresponds to the output variables, i.e. the pixels, and $\mathcal{E}$ includes the pairs of neighboring pixels.

Assume probability densities p_b and p_f estimated for the background and the foreground, respectively. This can be achieved, for example, by making use of the `EM algorithm`. The **unary energies** E_i for all $i \in \mathcal{V}$ can be defined as

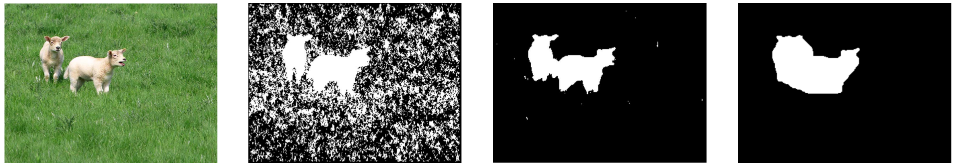
$$E_i(0, x_i) = 0$$
$$E_i(1, x_i) = \log \frac{p_b(x_i)}{p_f(x_i)} \; .$$

Contrast-sensitive Potts-model

The pairwise energy functions are defined as

$$E_{ij}(y_i, y_j; x_i, x_j) = w \exp(-\gamma \|x_i - x_j\|^2) [\![y_i \neq y_j]\!] \ ,$$

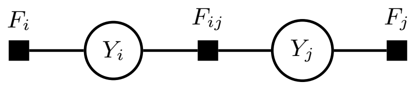
where $w \geqslant 0$ is a weighting factor. The parameter γ is the mean edge strength.



Original image | w is small | w is medium | w is high

Energy minimization via minimum $s - t$ cut

Let us consider the following example



Through this example we illustrate how to minimize *regular energy functions* consisting of up to pairwise relationships. In our example $\mathbf{y} \in \{0, 1\}^2$ and $E(\mathbf{y})$ is defined as

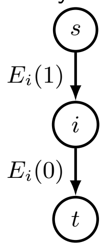
$$E(\mathbf{y}) = E_1(y_1) + E_2(y_2) + E_{12}(y_1, y_2) \ .$$

We will create a flow network $(\mathcal{V} \cup \{s, t\}, \mathcal{E}', c, s, t)$ such that the minimum $s - t$ cut will correspond to the minimization of our energy function $E(\mathbf{y})$, where the labeling for each $i \in \mathcal{V}$ is given by

$$y_i = \begin{cases} 0, & \text{if } i \in \mathcal{S} \ , \\ 1, & \text{if } i \in \mathcal{T} \ . \end{cases}$$

Graph construction: unary energies

Let us consider the unary energy function $E_i : \{0,1\} \to \mathbb{R}$.

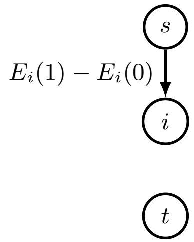


Obviously, the minimum $s-t$ cut of the flow network will correspond to

$$\underset{y_i \in \{0,1\}}{\operatorname{argmin}} E_i(y_i) \ .$$

Without loss of generality we can assume that $E_i(1) > E_i(0)$, then we can write

$$\underset{y_i \in \{0,1\}}{\operatorname{argmin}} E_i(y_i) = \underset{y_i \in \{0,1\}}{\operatorname{argmin}} E_i(y_i) - E_i(0) \ .$$



Graph construction: pairwise energies

Let us consider the pairwise energy function $E_{ij}(y_i, y_j) : \{0,1\}^2 \to \mathbb{R}$. The possible values of $E_{ij}(y_i, y_j)$ are shown in the table:

E_{ij}	$y_j = 0$	$y_j = 1$
$y_i = 0$	A	B
$y_i = 1$	C	D

We furthermore assume that $E_{ij}(y_i, y_j)$ is regular, that is

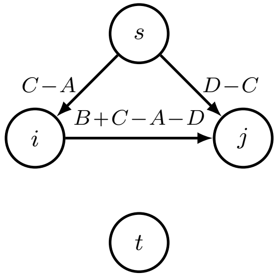
$$E_{ij}(0,0) + E_{ij}(1,1) \leqslant E_{ij}(0,1) + E_{ij}(1,0)$$
$$A + D \leqslant B + C \;.$$

Let us note that $E_{ij}(y_i, y_j)$ can be decomposed as:

$$\begin{array}{|c|c|}\hline A & B \\ \hline C & D \\ \hline \end{array} = A + \begin{array}{|c|c|}\hline 0 & B-A \\ \hline C-A & D-A \\ \hline \end{array} = A + \begin{array}{|c|c|}\hline 0 & 0 \\ \hline C-A & C-A \\ \hline \end{array} + \begin{array}{|c|c|}\hline 0 & B-A \\ \hline 0 & D-C \\ \hline \end{array}$$

$$= A + \underbrace{\begin{array}{|c|c|}\hline 0 & 0 \\ \hline C-A & C-A \\ \hline \end{array}}_{E_i(1)} + \underbrace{\begin{array}{|c|c|}\hline 0 & D-C \\ \hline 0 & D-C \\ \hline \end{array}}_{E_j(1)} + \underbrace{\begin{array}{|c|c|}\hline 0 & B+C-A-D \\ \hline 0 & 0 \\ \hline \end{array}}_{B+C-A-D \geqslant 0}$$

Graph construction: pairwise energies, $C - A \geqslant 0$, $D - C \geqslant 0$ *

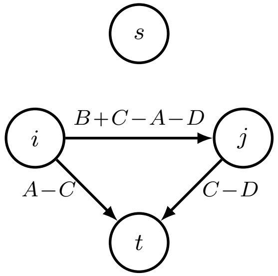


E_{ij}	$y_j = 0$	$y_j = 1$
$y_i = 0$	A	B
$y_i = 1$	C	D

Labeling: $y_i = y_j = 0$.

$$C - A \geqslant 0 \;\Rightarrow C \geqslant A\,.$$
$$D - C \geqslant 0 \;\Rightarrow D \geqslant C \;\Rightarrow\; D \geqslant A\,.$$
$$0 \leqslant B + C - A - D \leqslant B - A \;\Rightarrow B \geqslant A\,.$$

Graph construction: pairwise energies, $A - C \geqslant 0$, $C - D \geqslant 0$ *

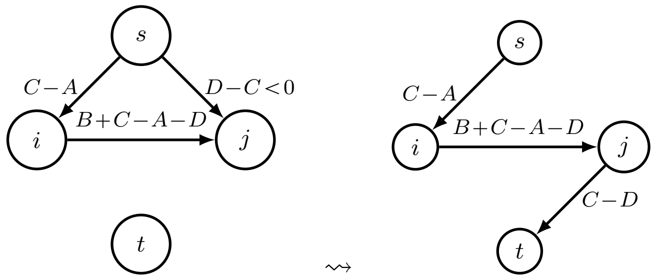


E_{ij}	$y_j = 0$	$y_j = 1$
$y_i = 0$	A	B
$y_i = 1$	C	D

Labeling: $y_i = y_j = 1$.

$$C - D \geqslant 0 \;\Rightarrow C \geqslant D\,.$$
$$A - C \geqslant 0 \;\Rightarrow A \geqslant C \;\Rightarrow\; A \geqslant D\,.$$
$$0 \leqslant B + C - A - D \leqslant B - D \;\Rightarrow B \geqslant D\,.$$

Graph construction: pairwise energies, $C - A \geqslant 0$, $C - D \geqslant 0$ *

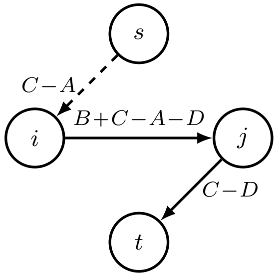


Note that the labeling $y_i = 1$, $y_j = 0$ in this case is not possible, since

$$C - A \geqslant 0 \;\Rightarrow\; C \geqslant A\,.$$

Graph construction: pairwise energies, $C - A \geqslant 0$, $C - D \geqslant 0$ *

Assume that $\min\{C - A, B + C - A - D, C - D\} = C - A$.



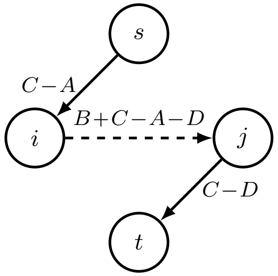
E_{ij}	$y_j = 0$	$y_j = 1$
$y_i = 0$	A	B
$y_i = 1$	C	D

Labeling: $y_i = y_j = 1$.

$$C - A \leqslant B + C - A - D \;\Rightarrow 0 \leqslant B - C \;\Rightarrow\; B \geqslant D\,.$$
$$C - A \geqslant C - D \;\Rightarrow A \geqslant D\,.$$
$$C - D \geqslant 0 \;\Rightarrow C \geqslant D\,.$$

Graph construction: pairwise energies, $C - A \geqslant 0$, $C - D \geqslant 0$ *

Assume that $\min\{C - A, B + C - A - D, C - D\} = B + C - A - D$.



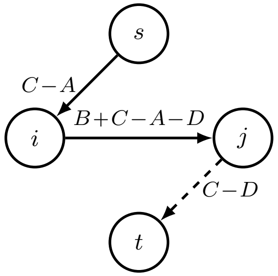
E_{ij}	$y_j = 0$	$y_j = 1$
$y_i = 0$	A	B
$y_i = 1$	C	D

Labeling: $y_i = 0$, $y_j = 1$.

$$B + C - A - D \leqslant C - A \;\Rightarrow B \leqslant D\,.$$
$$B + C - A - D \geqslant C - D \;\Rightarrow B \leqslant A\,.$$
$$C - A \geqslant 0 \;\Rightarrow A \leqslant C \;\Rightarrow\; B \leqslant C\,.$$

Graph construction: pairwise energies, $C - A \geqslant 0$, $C - D \geqslant 0$ *

Assume that $\min\{C - A, B + C - A - D, C - D\} = C - D$.

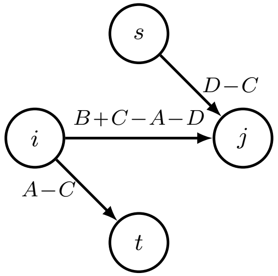


E_{ij}	$y_j = 0$	$y_j = 1$
$y_i = 0$	A	B
$y_i = 1$	C	D

Labeling: $y_i = y_j = 0$.

$$C - D \leqslant B + C - A - D \;\Rightarrow B \geqslant A\,.$$
$$C - D \leqslant C - A \;\Rightarrow D \geqslant A\,.$$
$$C - D \geqslant 0 \;\Rightarrow C \geqslant D \;\Rightarrow\; C \geqslant A\,.$$

Graph construction: pairwise energies, $A - C \geqslant 0$, $D - C \geqslant 0$ *



E_{ij}	$y_j = 0$	$y_j = 1$
$y_i = 0$	A	B
$y_i = 1$	C	D

Labeling: $y_i = 1$, $y_j = 0$.

$$D - C \geqslant 0 \;\Rightarrow D \geqslant C \;.$$

$$A - C \geqslant 0 \;\Rightarrow A \geqslant C \;.$$

$$0 \leqslant B + C - A - D \leqslant B - A \;\Rightarrow B \geqslant A \;\Rightarrow\; B \geqslant C \;.$$

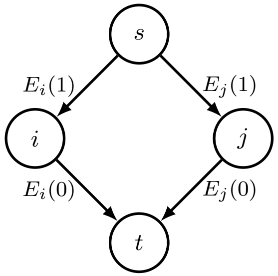
All the other cases can be similarly derived.

Graph construction

Putting all together we get that

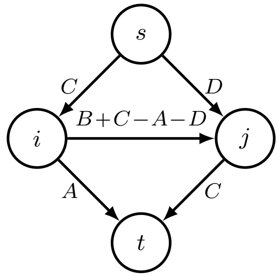
Unaries

$$\underset{\mathbf{y}}{\operatorname{argmin}}\, E_i(y_i) + E_j(y_j)$$



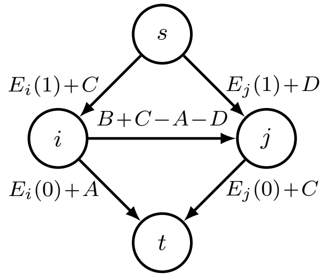
Pairwise

$$\underset{\mathbf{y}}{\operatorname{argmin}}\, E_{ij}(y_i, y_j)$$



Overall energy

$$\underset{\mathbf{y}}{\operatorname{argmin}}\, E_i(y_i) + E_j(y_j) + E_{ij}(y_i, y_j)$$



Remarks

Regularity is an *extremely important* property as is allows to minimize energy functions by making use of graph cut. Moreover, without the regularity constraint, the problem become intractable.

Let E_2 be a nonregular function of two binary variables. Then minimizing the energy function

$$E(y_1, \dots, y_n) = \sum_i E_i(y_i) + \sum_{i<j} E_2(y_i, y_j) ,$$

where E_i are arbitrary functions of one binary variable, is NP-hard.

Multi-label problem

Multi-label problem

We define a label set $\mathcal{L} = \{1, 2, \dots, L\}$, where L is a (finite) constant. Therefore the output domain is defined as $\mathcal{Y} = \mathcal{L}^{\mathcal{V}}$. The *energy function* has the following form

$$E(\mathbf{y}; \mathbf{x}) = \sum_{i \in \mathcal{V}} E_i(y_i; \mathbf{x}) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(y_i, y_j; \mathbf{x}) ,$$

where $\mathbf{x}$ consists of an input image.

In order to ease to notation we will omit $\mathbf{x}$ and define the *energy function* simply as

$$E(\mathbf{y}) = \sum_{i \in \mathcal{V}} E_i(y_i) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(y_i, y_j) .$$

Metric *

A function $d : \mathcal{L} \times \mathcal{L} \to \mathbb{R}^+$ is called a **metric** if the following properties are satisfied:

1. **Identity of indiscernibles**: $d(\ell_1, \ell_2) = 0 \quad \Leftrightarrow \quad \ell_1 = \ell_2$ for all $\ell_1, \ell_2 \in \mathcal{L}$.
2. **Symmetry**: $d(\ell_1, \ell_2) = d(\ell_2, \ell_1)$ for all $\ell_1, \ell_2 \in \mathcal{L}$.
3. **Triangle inequality**: $d(\ell_1, \ell_3) \leqslant d(\ell_1, \ell_2) + d(\ell_2, \ell_3)$ for all $\ell_1, \ell_2, \ell_3 \in \mathcal{L}$.

Example: the **truncated absolute distance** $d : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, $d(x, y) = \min(K, |x - y|)$ is a *metric*, where K is some constant. (See Exercise)

If a function $d : \mathcal{L} \times \mathcal{L} \to \mathbb{R}$ satisfies the first two properties (i.e. identity of indiscernibles and symmetric), then it is called **semi-metric**.

Example: the **truncated quadratic function** $d : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, $d(x, y) = \min(K, |x - y|^2)$ is a *semi-metric*, where K is some constant. (See Exercise)

$\alpha - \beta$ swap

$\alpha - \beta$ **swap** changes the variables that are labeled as $\ell \in \{\alpha, \beta\}$. Each of these variables can choose either α or β. We introduce the following notation

$$\mathcal{Z}_{\alpha\beta}(\mathbf{y}, \alpha, \beta) = \{\mathbf{z} \in \mathcal{Y} : z_i = y_i, \text{ if } y_i \notin \{\alpha, \beta\}, \text{ otherwise } z_i \in \{\alpha, \beta\}\} .$$

The minimization of the *energy function* E can be reformulated as follows:

$$\begin{aligned}
\hat{\mathbf{z}} \in & \operatorname*{argmin}_{\mathbf{z} \in \mathcal{Z}_{\alpha\beta}(\mathbf{y},\alpha,\beta)} E(\mathbf{z}) = \operatorname*{argmin}_{\mathbf{z} \in \mathcal{Z}_{\alpha\beta}(\mathbf{y},\alpha,\beta)} \sum_{i \in \mathcal{V}} E_i(z_i) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(z_i, z_j) \\
& = \operatorname*{argmin}_{\mathbf{z} \in \mathcal{Z}_{\alpha\beta}(\mathbf{y},\alpha,\beta)} \Bigg[\underbrace{\sum_{i \in \mathcal{V},\, y_i \notin \{\alpha,\beta\}} E_i(y_i)}_{\text{constant}} + \underbrace{\sum_{i \in \mathcal{V},\, y_i \in \{\alpha,\beta\}} E_i(z_i)}_{\text{unary}} \\
& \quad + \underbrace{\sum_{\substack{(i,j) \in \mathcal{E} \\ y_i, y_j \notin \{\alpha,\beta\}}} E_{ij}(y_i, y_j)}_{\text{constant}} + \underbrace{\sum_{\substack{(i,j) \in \mathcal{E} \\ y_i \in \{\alpha,\beta\},\, y_j \notin \{\alpha,\beta\}}} E_{ij}(z_i, y_j)}_{\text{unary}} + \underbrace{\sum_{\substack{(i,j) \in \mathcal{E} \\ y_i \notin \{\alpha,\beta\},\, y_j \in \{\alpha,\beta\}}} E_{ij}(y_i, z_j)}_{\text{unary}} + \underbrace{\sum_{\substack{(i,j) \in \mathcal{E} \\ y_i, y_j \in \{\alpha,\beta\}}} E_{ij}(z_i, z_j)}_{\text{pairwise}} \Bigg].
\end{aligned}$$

Local optimization

Let us consider $E_{ij}(z_i, z_j)$ for a given $(i,j) \in \mathcal{E}$:

E_{ij}	α	β
α	$E_{ij}(\alpha,\alpha)$	$E_{ij}(\alpha,\beta)$
β	$E_{ij}(\beta,\alpha)$	$E_{ij}(\beta,\beta)$

If we assume that $E_{ij} : \mathcal{L} \times \mathcal{L} \to \mathbb{R}$ is a **semi-metric** for each $(i,j) \in \mathcal{E}$, then

$$E_{ij}(\alpha,\alpha) + E_{ij}(\beta,\beta) = 0 \leqslant E_{ij}(\alpha,\beta) + E_{ij}(\beta,\alpha) = 2E_{ij}(\alpha,\beta) \ ,$$

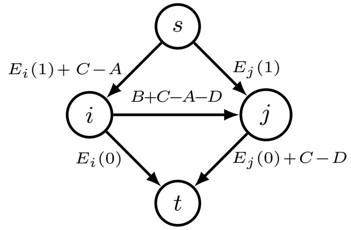
which means that E_{ij} is **regular** w.r.t. the labeling $\mathcal{Z}_{\alpha\beta}(\mathbf{y}, \alpha, \beta)$.

Graph construction for semi-metrics

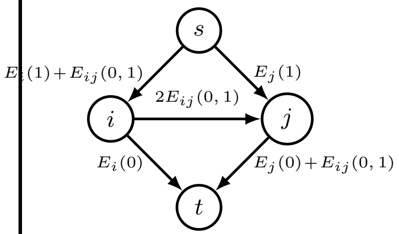
Let us consider the following *binary energy function*:

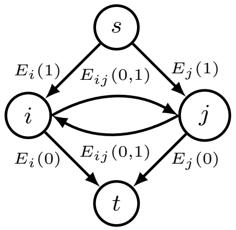
$$E(\mathbf{y}) = E_i(y_i) + E_j(y_j) + E_{ij}(y_i, y_j) \ ,$$

where E_{ij} is assumed to be a *semi-metric*.



Since E_{ij} is a *semi-metric*, we can construct a flow for $E(\mathbf{y})$ as follows:





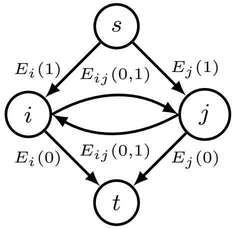
y_i	y_j	$E(\mathbf{y})$
0	0	$E_i(0)+E_j(0)$
0	1	$E_i(0)+E_j(1)+E_{ij}(1,0)$
1	0	$E_i(1)+E_j(0)+E_{ij}(1,0)$
1	1	$E_i(1)+E_j(1)$

Graph construction: t-links

We need to minimize the following **regular** *energy function*:

$$\hat{\mathbf{z}} \in \operatorname*{argmin}_{\mathbf{z}\in\mathcal{Z}_{\alpha\beta}(\mathbf{y},\alpha,\beta)} \sum_{\substack{i\in\mathcal{V}\\ y_i\in\{\alpha,\beta\}}} E_i(z_i) + \sum_{\substack{(i,j)\in\mathcal{E}\\ y_i\in\{\alpha,\beta\},\, y_j\notin\{\alpha,\beta\}}} E_{ij}(z_i,y_j) + \sum_{\substack{(i,j)\in\mathcal{E}\\ y_i\notin\{\alpha,\beta\},\, y_j\in\{\alpha,\beta\}}} E_{ij}(y_i,z_j) + \sum_{\substack{(i,j)\in\mathcal{E}\\ y_i,y_j\in\{\alpha,\beta\}}} E_{ij}(z_i,z_j).$$

Based on construction applied for *binary image segmentation*, we can also define a *flow network* $(\mathcal{V}',\mathcal{E}',c,\alpha,\beta)$, where $\mathcal{V}'=\{\alpha,\beta\}\cup\{i\in\mathcal{V}: y_i\in\{\alpha,\beta\}\}$ and $\mathcal{E}'=\underbrace{\{(\alpha,i),(i,\beta): i\in\mathcal{V}'\backslash\{\alpha,\beta\}\}}_{\text{t-links}}\cup\underbrace{\{(i,j),(j,i)\mid i,j\in\mathcal{V}'\backslash\{\alpha,\beta\},(i,j)\in\mathcal{E}\}}_{\text{n-links}}$.

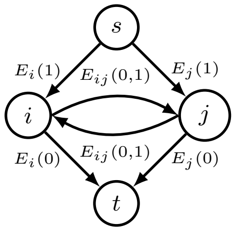


t-links: for all $i\in\mathcal{V}'\backslash\{\alpha,\beta\}$

$$c(\alpha,i) = E_i(\beta) + \sum_{(i,j)\in\mathcal{E},\, y_j\notin\{\alpha,\beta\}} E_{ij}(\beta,y_j) + \sum_{(j,i)\in\mathcal{E},\, y_j\notin\{\alpha,\beta\}} E_{ji}(y_j,\beta)\ .$$

$$c(i,\beta) = E_i(\alpha) + \sum_{(i,j)\in\mathcal{E},\, y_j\notin\{\alpha,\beta\}} E_{ij}(\alpha,y_j) + \sum_{(j,i)\in\mathcal{E},\, y_j\notin\{\alpha,\beta\}} E_{ji}(y_j,\alpha)\ .$$

Graph construction: n-links



n-links: for all $(i,j)\in\mathcal{E}$, where $i,j\in\mathcal{V}'\backslash\{\alpha,\beta\}$

$$c(i,j)=c(j,i)=E_{ij}(\alpha,\beta)\ .$$

$\alpha - \beta$ swap algorithm $*$

Input: An *energy function* $E(\mathbf{y}) = \sum_{i \in \mathcal{V}} E_i(y_i) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(y_i, y_j)$ to be minimized, where E_{ij} is a **semi-metric** for each $(i,j) \in \mathcal{E}$
Output: A local minimum $\mathbf{y} \in \mathcal{Y} = \mathcal{L}^{\mathcal{V}}$ of $E(\mathbf{y})$

```
1: Choose an arbitrary initial labeling y ∈ 𝒴
2: ŷ ← y
3: for all (α, β) ∈ ℒ × ℒ do
4:     find ẑ ∈ argmin_{z ∈ 𝒵_{αβ}(ŷ, α, β)} E(z)
5:     ŷ ← ẑ
6: end for
7: if E(ŷ) < E(y) then
8:     y ← ŷ
9:     Goto Step 2
10: end if
```

$\alpha - \beta$ swap algorithm is guaranteed to terminate in a finite number of cycles. This algorithm computes at least $|\mathcal{L}|^2$ graph cuts, which may take a lot of time, even for moderately large label spaces.

α-expansion

α-expansion allows each variable either to keep its current label or to change it to the label $\alpha \in \mathcal{L}$. We introduce the following notation

$$\mathcal{Z}_\alpha(\mathbf{y}, \alpha) = \{\mathbf{z} \in \mathcal{Y} : z_i \in \{y_i, \alpha\} \text{ for all } i \in \mathcal{V}\} .$$

The minimization of the *energy function* E can be reformulated as follows:

$$\begin{aligned}
\hat{\mathbf{z}} \in \operatorname*{argmin}_{\mathbf{z} \in \mathcal{Z}_\alpha(\mathbf{y},\alpha)} E(\mathbf{z}) &= \operatorname*{argmin}_{\mathbf{z} \in \mathcal{Z}_\alpha(\mathbf{y},\alpha)} \sum_{i \in \mathcal{V}} E_i(z_i) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(z_i, z_j) \\
&= \operatorname*{argmin}_{\mathbf{z} \in \mathcal{Z}_\alpha(\mathbf{y},\alpha)} \Bigg[\underbrace{\sum_{i \in \mathcal{V},\, y_i = \alpha} E_i(\alpha)}_{\text{constant}} + \underbrace{\sum_{i \in \mathcal{V},\, y_i \neq \alpha} E_i(z_i)}_{\text{unary}} \\
&\quad + \underbrace{\sum_{\substack{(i,j) \in \mathcal{E} \\ y_i = \alpha,\, y_j = \alpha}} E_{ij}(\alpha, \alpha)}_{\text{constant}} + \underbrace{\sum_{\substack{(i,j) \in \mathcal{E} \\ y_i = \alpha,\, y_j \neq \alpha}} E_{ij}(\alpha, z_j)}_{\text{unary}} + \underbrace{\sum_{\substack{(i,j) \in \mathcal{E} \\ y_i \neq \alpha,\, y_j = \alpha}} E_{ij}(z_i, \alpha)}_{\text{unary}} + \underbrace{\sum_{\substack{(i,j) \in \mathcal{E} \\ y_i \neq \alpha,\, y_j \neq \alpha}} E_{ij}(z_i, z_j)}_{\text{pairwise}} \Bigg] .
\end{aligned}$$

Local optimization

Let us consider $E_{ij}(z_i, z_j)$ for a given $(i,j) \in \mathcal{E}$:

E_{ij}	α	y_j
α	$E_{ij}(\alpha,\alpha)$	$E_{ij}(\alpha,y_j)$
y_i	$E_{ij}(y_i,\alpha)$	$E_{ij}(y_i,y_j)$

If we assume that $E_{ij} : \mathcal{L} \times \mathcal{L} \to \mathbb{R}$ is a **metric** for each $(i,j) \in \mathcal{E}$, then

$$E_{ij}(\alpha,\alpha) + E_{ij}(y_i,y_j) = E_{ij}(y_i,y_j) \leqslant E_{ij}(y_i,\alpha) + E_{ij}(\alpha,y_j) ,$$

which means that E_{ij} is **regular** w.r.t. the labeling $\mathcal{Z}_\alpha(\mathbf{y},\alpha)$.

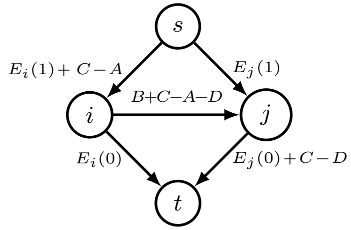
Graph construction

We need to minimize the following **regular** *energy function*:

$$\hat{\mathbf{z}} \in \underset{\mathbf{z}\in\mathcal{Z}_\alpha(\mathbf{y},\alpha)}{\operatorname{argmin}} \underbrace{\sum_{\substack{i\in\mathcal{V}\\ y_i\neq\alpha}} E_i(z_i) + \sum_{\substack{(i,j)\in\mathcal{E}\\ y_i=\alpha,\, y_j\neq\alpha}} E_{ij}(\alpha, z_j) + \sum_{\substack{(i,j)\in\mathcal{E}\\ y_i\neq\alpha,\, y_j=\alpha}} E_{ij}(z_i,\alpha)}_{\text{unary}} + \underbrace{\sum_{\substack{(i,j)\in\mathcal{E}\\ y_i\neq\alpha,\, y_j\neq\alpha}} E_{ij}(z_i,z_j)}_{\text{pairwise}} .$$

Based on construction applied for *binary image segmentation*, we can also define a *flow network* $(\mathcal{V}', \mathcal{E}', c, \alpha, \bar{\alpha})$, where $\mathcal{V}' = \{\alpha, \bar{\alpha}\} \cup \{i \in \mathcal{V} : y_i \neq \alpha\}$ and $\mathcal{E}' = \underbrace{\{(\alpha,i),(i,\bar{\alpha}) : i \in \mathcal{V}' \backslash \{\alpha,\bar{\alpha}\}\}}_{\text{t-links}} \cup \underbrace{\{(i,j) \in \mathcal{E} : i,j \in \mathcal{V}' \backslash \{\alpha,\bar{\alpha}\}\}}_{\text{n-links}}$.

Graph construction: t-links

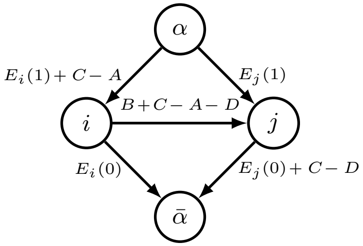


t-links: for all $i \in \mathcal{V}' \backslash \{\alpha, \bar{\alpha}\}$

$$c(\alpha, i) = E_i(y_i) + \sum_{(i,j)\in\mathcal{E},\, y_j = \alpha} E_{ij}(y_i, \alpha) + \sum_{(j,i)\in\mathcal{E},\, y_j = \alpha} E_{ji}(\alpha, y_i) + \underbrace{\sum_{(i,j)\in\mathcal{E},\, y_j \neq \alpha} E_{ij}(y_i, \alpha)}_{C} \; .$$

$$c(i, \bar{\alpha}) = E_i(\alpha) + \underbrace{\sum_{(j,i)\in\mathcal{E},\, y_j \neq \alpha} E_{ji}(y_j, \alpha)}_{C} - \underbrace{\sum_{(j,i)\in\mathcal{E},\, y_j \neq \alpha} E_{ji}(y_j, y_i)}_{D} \; .$$

Graph construction: n-links



n-links: for all $(i, j) \in \mathcal{E}$, where $i, j \in \mathcal{V}' \backslash \{\alpha, \bar{\alpha}\}$

$$c(i, j) = E_{ij}(\alpha, y_j) + E_{ij}(y_i, \alpha) - E_{ij}(y_i, y_j) \; .$$

α-expansion algorithm *

Input: An *energy function* $E(\mathbf{y}) = \sum_{i \in \mathcal{V}} E_i(y_i) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(y_i, y_j)$ to be minimized, where E_{ij} is a **metric** for each $(i,j) \in \mathcal{E}$
Output: A local minimum $\mathbf{y} \in \mathcal{Y} = \mathcal{L}^{\mathcal{V}}$ of $E(\mathbf{y})$

1: Choose an arbitrary initial labeling $\mathbf{y} \in \mathcal{Y}$
2: $\hat{\mathbf{y}} \leftarrow \mathbf{y}$
3: **for all** $\alpha \in \mathcal{L}$ **do**
4:     find $\hat{\mathbf{z}} \in \operatorname{argmin}_{\mathbf{z} \in \mathcal{Z}_\alpha(\hat{\mathbf{y}}, \alpha)} E(\mathbf{z})$
5:     $\hat{\mathbf{y}} \leftarrow \hat{\mathbf{z}}$
6: **end for**
7: **if** $E(\hat{\mathbf{y}}) < E(\mathbf{y})$ **then**
8:     $\mathbf{y} \leftarrow \hat{\mathbf{y}}$
9:     Goto Step 2
10: **end if**

α-expansion is guaranteed to terminate in a finite number of cycles. This algorithm computes at least $|\mathcal{L}|$ graph cuts, which may take a lot of time, when the label space $\mathcal{L}$ is large.

Optimality *

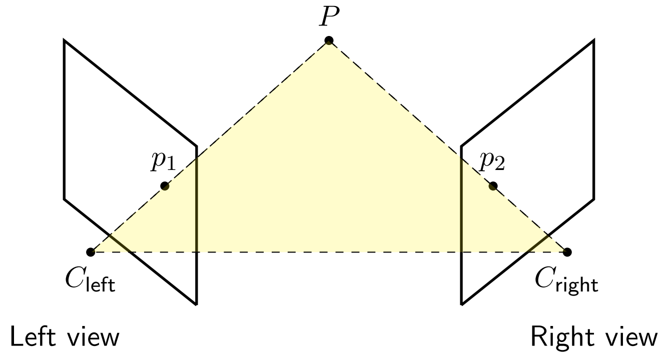
The $\alpha - \beta$ swap does not guarantee any closeness to the global minimum. Nevertheless, the local minimum that the α-expansion algorithm will find is at most twice the global minimum $\mathbf{y}^*$.

We have already assumed that E_{ij} is a metric for each $(i,j) \in \mathcal{E}$, hence $E_{ij}(\alpha, \beta) \neq 0$ for $\alpha \neq \beta \in \mathcal{L}$. Let us define

$$c = \max_{(i,j) \in \mathcal{E}} \left(\frac{\max_{\alpha \neq \beta \in \mathcal{L}} E_{ij}(\alpha, \beta)}{\min_{\alpha \neq \beta \in \mathcal{L}} E_{ij}(\alpha, \beta)} \right) .$$

Theorem 1. *Let $\hat{\mathbf{y}}$ be a local minimum when the expansion moves are allowed and $\mathbf{y}^*$ be the globally optimal solution. Then $E(\hat{\mathbf{y}}) \leqslant 2cE(\mathbf{y}^*)$.*

Stereo matching

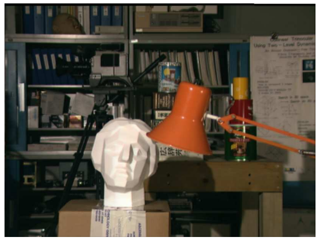


Given two images (i.e. left and right), two observed 2D points p_1 and p_2 on the *left image* and *right image*, respectively, corresponding to a 3D point P in $\mathbb{R}^3$. Note that P can be determined based on p_1 and p_2.
For more details you may refer to the course **Computer Vision II: Multiple View Geometry (IN2228)**.

Stereo matching

The goal is to reconstruct 3D points according to corresponding pixels.
We assume **rectified images** (i.e. the directions of the cameras are parallel), which means that the corresponding pixels are situated in **horizontal lines** according to some displacement.

Left view



Right view

Therefore, we need to search for corresponding points in the same row of both views. We also assume that the pixels p_1 and p_2 corresponding to P have similar intensities.

Energy function

We define $\mathcal{L} = \{1, 2, \ldots, D\}$ as the **label set**, i.e. set of possible *horizontal displacement* of pixels on the *right view*), where D is a constant. Therefore the output domain $\mathcal{Y} = \mathcal{L}^{\mathcal{V}}$ and the *energy function* has the following form

$$E(\mathbf{y}; \mathbf{x}) = \sum_{i \in \mathcal{V}} E_i(y_i; \mathbf{x}) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(y_i, y_j; \mathbf{x}) \ ,$$

where $\mathbf{x}$ consists of the images (i.e. left and right view) denoted by $\mathbf{x}^{\text{left}}$ and $\mathbf{x}^{\text{right}}$, respectively.

Unary energies (a.k.a. **data terms**) E_i for all $i \in \mathcal{V}$ are defined as

$$E_i(y_i; \mathbf{x}) = \min(|x_i^{\text{left}} - x_{i+y_i}^{\text{right}}|^2, K)) \ ,$$

where K is a constant (e.g., $K = 20^2$).

Energy function

Pairwise energies (a.k.a. **smooth terms**) E_{ij} for all $(i, j) \in \mathcal{E}$ are defined as

$$E_{ij}(y_i, y_j; \mathbf{x}) = U(|x_i^{\text{left}} - x_j^{\text{left}}|) \cdot [\![y_i \neq y_j]\!] \ ,$$

where

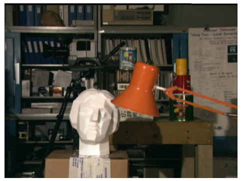
$$U(|x_i^{\text{left}} - x_j^{\text{left}}|) = \begin{cases} 2C, & \text{if } |x_i^{\text{left}} - x_j^{\text{left}}| \leqslant 5 \\ C, & \text{otherwise} \end{cases}$$

for some constant C.

Note the pairwise energies are defined by **weighted Potts-model**, which is a metric (see Exercise).

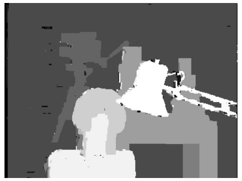
Results *

Left view

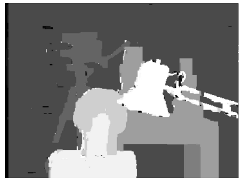


Right view

Ground truth



Result of $\alpha - \beta$ swap



Result of α-expansion

It is worth noting that α-expansion algorithm generally runs faster than $\alpha - \beta$ swap. There is optimality guarantee only for α-expansion algorithm, however, the two algorithms perform almost the same in many practical applications.

Summary *

- A binary energy function E consisting of up to pairwise functions is **regular**, if for each term E_{ij} for all $i < j$ satisfies
$$E_{ij}(0,0) + E_{ij}(1,1) \leqslant E_{ij}(0,1) + E_{ij}(1,0) \ .$$
- The *minimization of regular energy functions* can be achieved via *graph cut*.
- The *multi-label problem* for a finite label set $\mathcal{L}$
$$E(\mathbf{y};\mathbf{x}) = \sum_{i\in\mathcal{V}} E_i(y_i;\mathbf{x}) + \sum_{(i,j)\in\mathcal{E}} E_{ij}(y_i,y_j;\mathbf{x}) \ ,$$
can be approximately solved by applying
  - $\alpha - \beta$ `swap`, if E_{ij} is semi-metric;
  - α-`expansion`, if E_{ij} is metric.

In the **next lecture** we will learn about

- Linear programming relaxation for multi-label problem
- `Fast primal-dual algorithm`

Literature *

1. Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, February 2004
2. Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001
3. Sebastian Nowozin and Christoph H. Lampert. Structured prediction and learning in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3–4), 2010