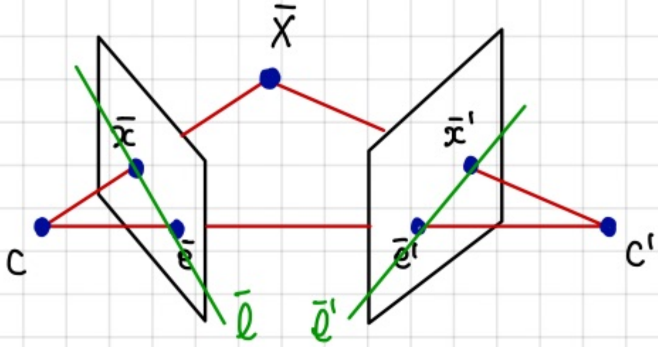
LECTURE 8 : DEPTH FROM STEREO

[1] Fundamental Matrix



C = the camera center
$\bar{e}$ = the epipole
$\bar{\ell}$ = the epipolar line

What are the possible correlations between $\bar{x}$ & $\bar{x}'$ above?

Homography

$$\bar{x}' = \mathbb{H}\bar{x}$$

$$\Downarrow$$

What is $\mathbb{F}$?

Fundamental Matrix

$$\boxed{\begin{array}{ll} (1) & \bar{x}'^{T}\mathbb{F}\bar{x} = 0 \\ (2) & \bar{\ell}' = \mathbb{F}\bar{x} \end{array}}$$

$\mathbb{F}$ is a 3x3 matrix with rank 2

$$\bar{\ell}' = \bar{x}' \times \bar{e}' \rightarrow$$

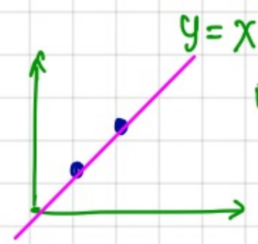
$$\bar{\ell}' = [e']_{\times}\bar{x}'$$

Two points create a line.

e.g.: $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \times \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \rightarrow$ line equation:

$\downarrow$ $\bar{\ell}$

$$ax + by + c = 0$$
$$-1x + 1y + 0 = 0$$
$$y = x$$



Skew-symmetric matrix:

$$[e']_{\times} = \begin{bmatrix} 0 & -e'_z & e'_y \\ e'_z & 0 & -e'_x \\ -e_y & e'_x & 0 \end{bmatrix}$$

$\downarrow$

Purpose:
to transform a cross product with a vector to a matrix multiplication

$$\bar{x}' = \mathbb{H}\bar{x}$$

$$\bar{\ell}' = [e']_{\times}\mathbb{H}\bar{x}$$

$$\boxed{\bar{\ell}' = \mathbb{F}\bar{x}} \quad ; \text{ where } \mathbb{F} = [e']_{\times}\mathbb{H}$$

Another look at $\mathbb{F}$:

$$\bar{x} = \mathbb{P}\bar{X}$$

Using the backward projection:

$$\bar{X} = \bar{C} + \lambda\, \mathbb{P}^{+}\bar{x}$$

→ This is a ray governed by two points: $\mathbb{P}^{+}\bar{x}$ & $\bar{C}$

Projecting the two points ($\mathbb{P}^{+}\bar{x}$ and $\bar{C}$) onto another image:

point $\mathbb{P}^{+}\bar{x} \longrightarrow \mathbb{P}'\mathbb{P}^{+}\bar{x}$

point $\bar{C} \longrightarrow \mathbb{P}'\bar{C}$ → $\mathbb{P}'\bar{C} = \bar{e}'$

The epipolar line of these two projected points:

$$\bar{\ell}' = (\mathbb{P}'\mathbb{P}^{+}\bar{x}) \times (\mathbb{P}'\bar{C}) = (\mathbb{P}'\mathbb{P}^{+}\bar{x}) \times \bar{e}'$$
$$= [\bar{e}']_{\times}\, \mathbb{P}'\mathbb{P}^{+}\bar{x}$$

Thus, we can also define: $\mathbb{F} = [\bar{e}']_{\times}\, \mathbb{P}'\mathbb{P}^{+}$

$$\boxed{\mathbb{F} = [\mathbb{P}'\bar{C}]_{\times}\, \mathbb{P}'\mathbb{P}^{+}}$$

From the last equation, we can infer that:

1. $\mathbb{F}$ is independent from the world structures (X), and depends only on the camera properties.
2. An image pair has a single value of $\mathbb{F}$.

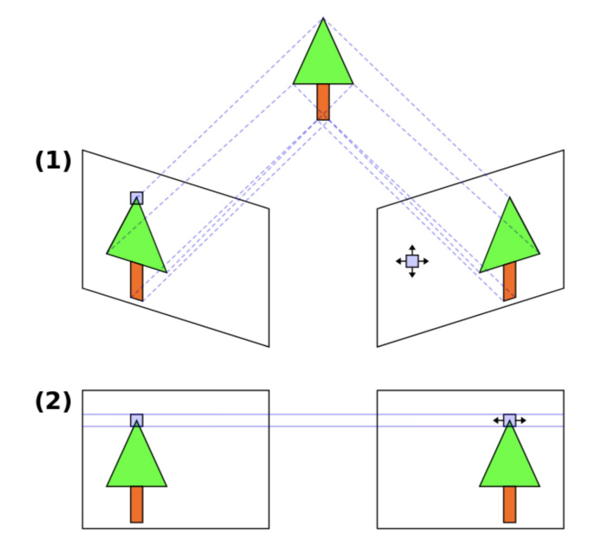
Properties of F

- F is a rank 2 homogeneous matrix with 7 degrees of freedom.
- **Point correspondence**: If $\mathbf{x}$ and $\mathbf{x}'$ are corresponding image points, then $\mathbf{x}'^{\top}\mathtt{F}\mathbf{x} = 0$.
- **Epipolar lines**:
  - ◇ $\mathbf{l}' = \mathtt{F}\mathbf{x}$ is the epipolar line corresponding to $\mathbf{x}$.
  - ◇ $\mathbf{l} = \mathtt{F}^{\top}\mathbf{x}'$ is the epipolar line corresponding to $\mathbf{x}'$.
- **Epipoles**:
  - ◇ $\mathtt{F}\mathbf{e} = \mathbf{0}$ $\qquad$ $\mathtt{F}^{\top}\mathbf{e}' = \mathbf{0}$
- **Computation from camera matrices** $\mathtt{P}, \mathtt{P}'$:
  - ◇ $\mathtt{F} = [\mathtt{P}'\mathbf{c}]_{\times}\mathtt{P}'\mathtt{P}^{+}$, where $\mathtt{P}^{+}$ is the pseudo-inverse of $\mathtt{P}$, and $\mathbf{c}$ is the centre of the first camera. Note, $\mathbf{e}' = \mathtt{P}'\mathbf{c}$.
  - ◇ Canonical cameras, $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}], \mathtt{P}' = [\mathtt{M} \mid \mathbf{m}]$,
    $\mathtt{F} = [\mathbf{e}']_{\times}\mathtt{M} = \mathtt{M}^{-\top}[\mathbf{e}]_{\times}$, where $\mathbf{e}' = \mathbf{m}$ and $\mathbf{e} = \mathtt{M}^{-1}\mathbf{m}$.

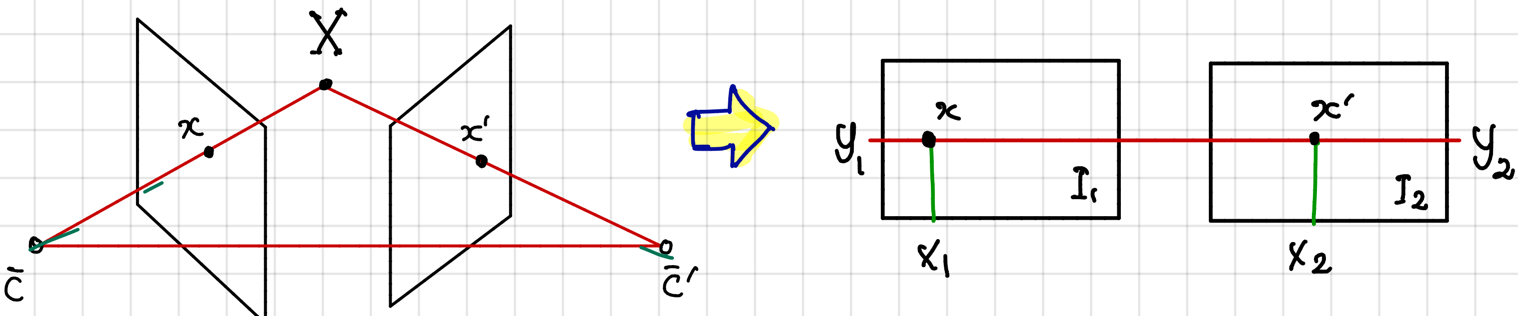
The lines are the epipolar lines corresponding to the points on the left figure.

[2] Stereo Image Rectification

The goal of image rectification:

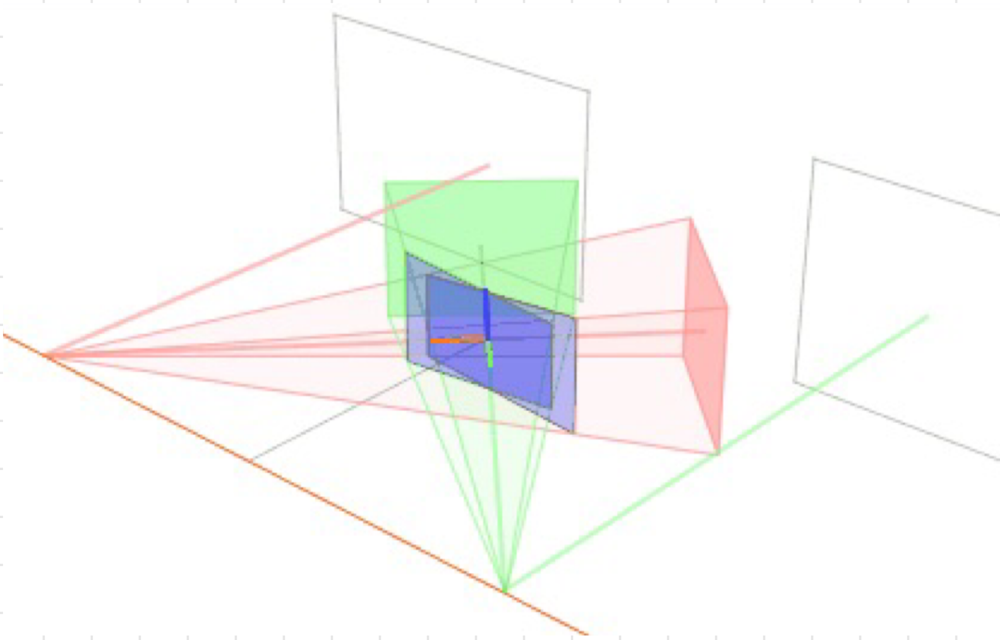


the search space before (1) and after (2) rectification



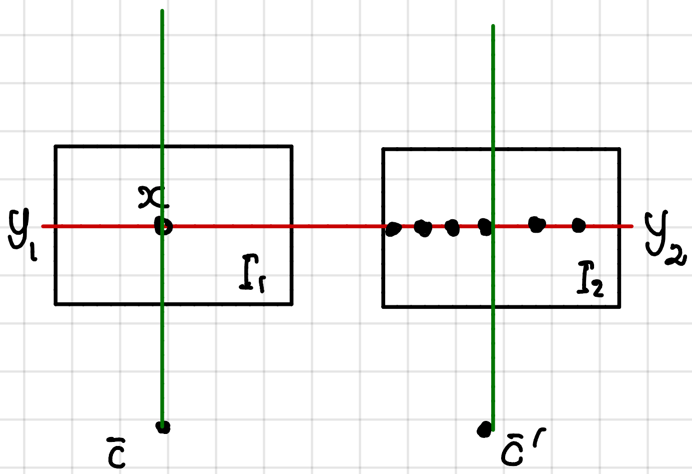
$$I_1(x) = I_2(x')$$

Image rectification process:



source: Wikipedia

[3] Disparity & Depth

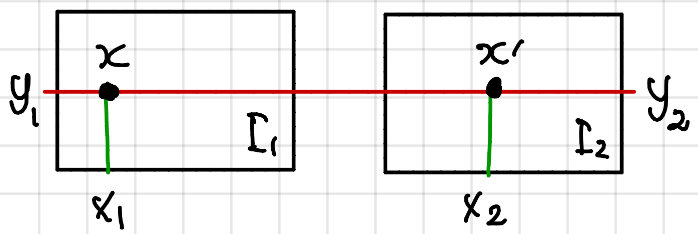


All the points on the red line on the right image are the candidates of a point, x', that make $I_2(x') = I_1(x)$

$\Downarrow$

where x & x' represents the same point in the world.

This red line is thus useful to find x' given x. Since, now we don't need to search x' in the entire image of I_2.



$$I_1(x) = I_2(x')$$

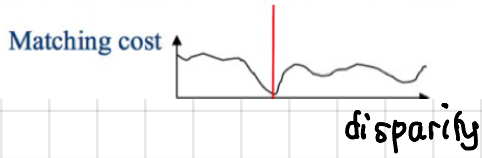
$$I_1(x_1, y_1) = I_2(x_2, y_2)$$

$$y_1 = y_2$$

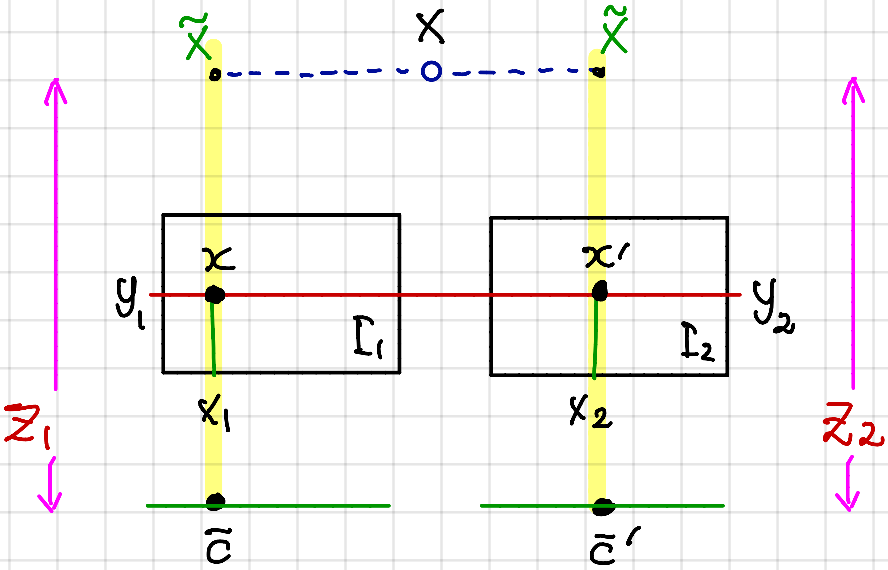
Disparity $\rightarrow$ $d = |x_1 - x_2|$

The larger the disparity, the smaller the depth : $d \propto 1/z$

left image

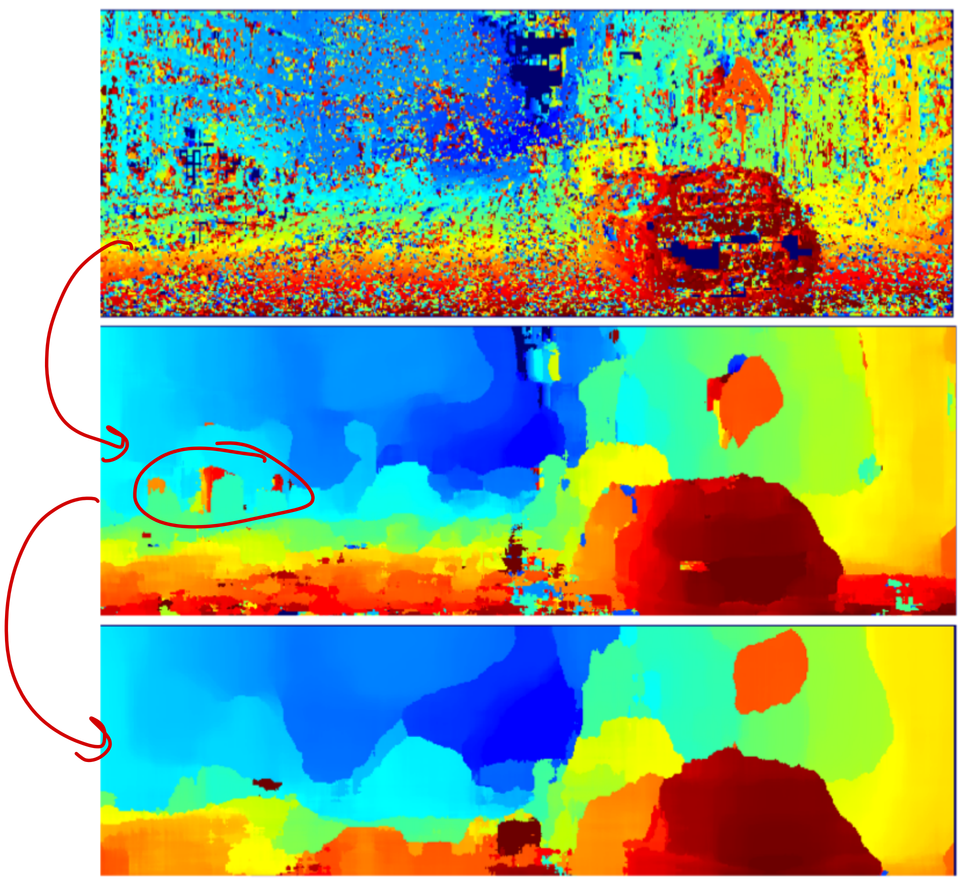


For a pair of rectified images, there is only one disparity or depth map:



[3] Disparity Map from Rectified Images

Pixel or patch based operations:



patch size $= 5$

patch size $= 35$

patch size $= 85$

Problem: small patches → more details but more noise
large patch → less noise but less details

Solution: MRF (Markov Random Fields)

[4] Markov Random Field

Markov Random Field (MRF) is one type of graphical models

⇓

Motivation : Graphical Models

① An image is a set of numbers representing light intensity / brightness and arranged orderly in a 2D space.

② To make sense those numbers, we need to find the correlations among the numbers.

③ To find the correlations means to formulize the numbers spatially.

④ One of the methods to formulize is through the graphical models

↓

A graphical model is a probabilistic model for which a graph expresses the conditional dependence structure between random variables.

↓

This requires us to understand the basic principles of probabilistic inference.

[5] Basic Probability Theory

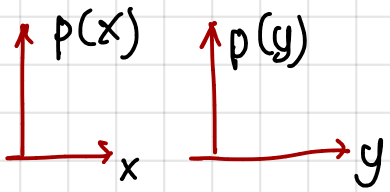
Product rule

Independent variables

$$p(x|y) = p(x)$$
$$p(y|x) = p(y)$$

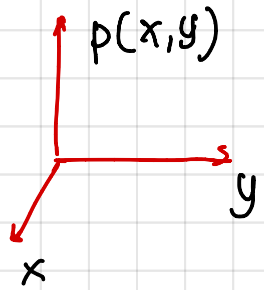
Thus:

$$p(x,y) = p(x)\,p(y)$$



Dependent variables

$$p(x,y) = p(x|y)\,p(y)$$
$$= p(y|x)\,p(x)$$



Notes:

① $0 \leqslant p(x) \leqslant 1$

② $\int p(x)\,dx = 1$

$\sum_x p(x) = 1$

Sum rule (Marginalization)

Two dependent variables x and y:

$$p(x) = \sum_y p(x,y)$$

Example:

x & y are random variables representing the possible faces of a coin

$x, y \in \{\text{Head}, \text{Tail}\}$

Then:

$$p(x=H) = \sum_{y \in \{H,T\}} p(x=H, y)$$
$$= \underbrace{p(x=H, y=H)}_{1/4} + \underbrace{p(x=H, y=T)}_{1/4}$$

Three dependent variables, x, y, z:

$$p(x) = \sum_y \sum_z p(x,y,z)$$

$$p(x,y) = \sum_z p(x,y,z)$$

[6] Bayes' Theorem

$$p(x,y) = p(x|y)\,p(y) \quad \text{(by product rule)}$$

$$p(x|y) = \frac{p(x,y)}{p(y)}$$

Thus:

$$\boxed{p(x|y) = \frac{p(y|x)\,p(x)}{p(y)}}$$

$$\text{Posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

$$\Downarrow$$

$$p(x|d) = \frac{p(d|x)\,p(x)}{p(d)}$$

where:

d = the observed data (the value known to us)

x = the hidden / random variable (the value to estimate)

$p(x|d)$ = the probability of x having a certain value given the data, d.

$p(d|x)$ = the likelihood probability of the data assuming x having a certain value.

$p(x)$ = the prior probability of x

$p(d)$ = the evidence, where:

$$p(d) = \sum_x p(d,x) \quad \text{(by sum rule)}$$

$$= \sum_x p(d|x)\,p(x)$$ → expensive to compute if the dimension of x is large.

$$p(d) = \int p(d|x)\,p(x)\,dx$$ → in most cases it's intractable to integrate.

[7] Probabilistic Inference Methods

Maximum Likelihood Estimation (MLE)

$$x^* = \underset{\{x\}}{\operatorname{argmax}}\ \underbrace{p(d|x)}_{\text{likelihood}}$$

x^* = the most probable value of x.

$\{x\}$ = a set of possible values of x (the candidates of x)

argmax = to return the value of x that maximizes $p(d|x)$

Maximum A Posteriori (MAP)

$$x^* = \operatorname{argmax}\ p(x|d)$$

$$= \operatorname{argmax}\ \underbrace{p(d|x)}_{\text{likehood}}\ \underbrace{p(x)}_{\text{prior}}$$

Full Bayesian Inference

$$p(x|d) = \frac{p(d|x)\ p(x)}{p(d)}$$

it's intractable to solve.

Example: According to our observation, a patient has a set of symptomps (= coughing, headache, fever), d.
We also have a set of possible illness: flu, infection and cancer.

The intuitive meaning of MLE:

1. Take each possible illness (e.g. x = flu)
2. Verify if it fits to the symptomps → probability of x.
3. Decide the illness based on the highest probability

[8] Maximum A Posteriori (MAP)

$$x^* = \underset{\{x\}}{\operatorname{argmax}}\ p(x|d) = \underset{\{x\}}{\operatorname{argmax}}\ \frac{p(d|x)\,p(x)}{p(d)}$$

$$x^* = \underset{\{x\}}{\operatorname{argmax}}\ p(d|x)\,p(x)$$

Because $p(d)$ is independent from x.

Notes:

1. The weakness of MLE is that it allows impossible candidates, as it treats candidates uniformly.
2. MAP treats the candidates differently: $p(x = flu) > p(x = cancer)$
3. Main criticism of MAP is how to determine $p(x)$.

[6] Full Bayesian Inference

$$p(x|d) = \frac{p(d|x)\,p(x)}{p(d)}$$

1. This is the ideal and most robust inference that besides providing a prediction on the value of x, it also provides the uncertainty, or how high/low the probability of the prediction.
2. It's possible because of $p(d)$, the evidence. This will normalize the product of $p(d|x)\,p(x)$.
3. However to obtain $p(d)$ is analytically intractable, and in discrete, it can be prohibitively expensive.

[9] Graphical Models: Factorization

Graphical model: A probabilistic model where a graph expresses the conditional dependence structure between random variables.

Conditional Indepedence:

$$p(x_1 | x_2, x_3, x_4) = p(x_1 | x_2)$$

↓

It implies that x_1 is independent of x_3 and x_4 given x_2.

↓

Conditional independence is the basic theory of the factorization of joint probability distribution.

Factorization of joint probability

↓

$$p(x_1, x_2, x_3, x_4) = p(x_3 | x_2, x_1) \, p(x_2 | x_1) \, p(x_1)$$

If x_3 is conditional independent of x_1, then:

$$p(x_1, x_2, x_3) = p(x_3 | x_2) \, p(x_2 | x_1) \, p(x_1)$$

↓

Factorization reduces the complexity of computing a joint probability

↓

Graphical model is useful due to factorization.

Computing $p(x_1, x_2, x_3)$ means we calculate all possible values of x_1, x_2, x_3. E.g. $x_i \in \{0, 1\}$

then:

$$\left.\begin{array}{l} x_1 = 0, x_2 = 0, x_3 = 0 \\ x_1 = 0, x_2 = 0, x_3 = 1 \\ x_1 = 0, x_2 = 1, x_3 = 0 \\ \vdots \end{array}\right\} 2^n$$

If we use factorization:

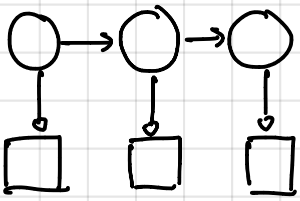
$$p(x_1, x_2, x_3) = p(x_1) \, p(x_2) \, p(x_3)$$

the computation is only: $2n$

[10] Types of Graphical Models

Directed Graphs:

e.g.:



Formulation:

$$p(x_1, \dots, x_N) = \prod_n p(x_n \mid x_{pa(x_n)})$$

where:

$x_{pa}(x_n)$ = the parents of x_n

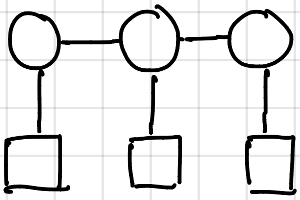
Example:

$$x_1 \to x_2 \to x_3$$

$$p(x_1, x_2, x_3) = p(x_3 \mid x_2)\, p(x_2 \mid x_1)\, p(x_1)$$

Undirected Graphs

e.g.:



Formulation:

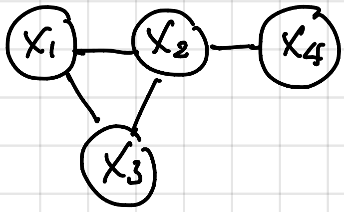
$$p(x_1 \dots x_N) = \frac{1}{Z} \prod_c \phi_c[x_1 \dots x_N]$$

where: c = a set of cliques.

Z = a normalization factor.

A clique is a set of mutually connected nodes.

e.g.:



Cliques: $\{\{x_1, x_2, x_3\}, \{x_2, x_4\}\}$

$\downarrow$

$$p(x_1, \dots x_4) = \frac{\phi[x_1, x_2, x_3]\, \phi[x_2, x_4]}{\sum_{x_1}\sum_{x_2}\sum_{x_3}\sum_{x_4} \phi[x_1, x_2, x_3]\, \phi[x_2, x_4]}$$

$$Z = \sum_{x_1}\sum_{x_2}\sum_{x_3}\sum_{x_4} \phi[x_1, x_2, x_3]\, \phi[x_2, x_4]$$

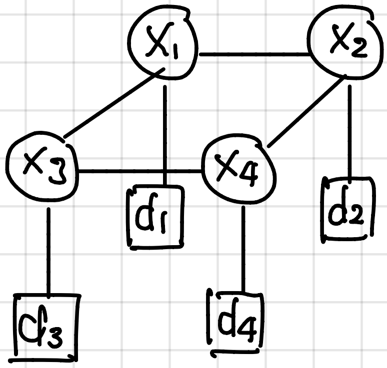
This Z is prohibitively exponsive to compute.

Most undirected graphs utilize MAP to ignore Z.

[11] MRF Formulation

A Markov random field (MRF) is a loopy undirected graph.

e.g. :



Based on the cliques:

$$
\begin{aligned}
p(\{x\},\{d\}) &= p(x_1 \dots x_4, d_1 \dots d_4) \\
&= \frac{1}{Z} \phi[x_1,d_1]\,\phi[x_2,d_2]\,\phi[x_3,d_3]\,\phi[x_4,d_4] \\
&\quad \phi[x_1,x_2]\,\phi[x_2,x_4]\,\phi[x_4,x_3]\,\phi[x_1,x_3] \\
&= \frac{1}{Z} \prod_{i}^{4} \phi[x_i,d_i] \prod_{j \in N_i} \phi[x_i,x_j]
\end{aligned}
$$

General MRF's formulation:

$$
\boxed{p(\{x\},\{d\}) = \frac{1}{Z} \prod_{i}^{N} \phi[x_i,d_i] \prod_{j \in N_i} \phi[x_i,x_j]}
$$

Using MAP:

$$
\begin{aligned}
\{x\}^* &= \underset{\{x\}}{\operatorname{argmax}}\; p(\{x\},\{d\}) \propto \underset{\{x\}}{\operatorname{argmax}}\; p(\{x\} \mid \{d\}) \\
&= \underset{\{x\}}{\operatorname{argmax}}\; \frac{1}{Z} \prod_{i} \phi[x_i,d_i] \prod_{j \in N_i} \phi[x_i,x_j] \\
&\propto \underset{\{x\}}{\operatorname{argmax}} \sum_{i} \left[\log \phi[x_i,d_i] + \sum_{j \in N_i} \log \phi[x_i,x_j] \right]
\end{aligned}
$$

$$
\boxed{\{x\}^* = \operatorname{argmin} \sum_{i} f_d(x_i,d_i) + \sum_{j \in N_i} f_p(x_i,x_j)}
$$

[12] MRFs : Data & Prior Terms

$$\{x\}^* = \arg\min \sum_i \underbrace{f_d(x_i, d_i)}_{\text{Data term}} + \sum_{j \in N_i} \underbrace{f_p(x_i, x_j)}_{\text{Prior term}}$$

Data term

The basic idea of determining the data term:

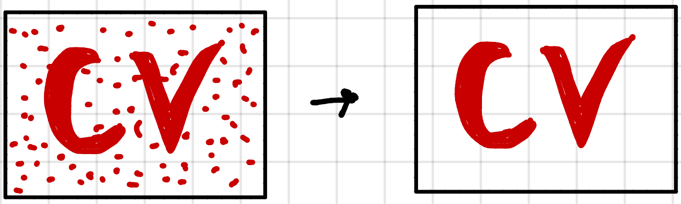
Due to argmin, $f_d(x_i, d_i)$ should produce a smaller value for a more probable x_i's value w.r.t. d_i.

Prior term:

The basic idea of determining the prior term (smoothness prior):

$f_p(x_i, x_j)$ should generate a smaller value when x_i & x_j have the same value.

Example: Denoising



1. $x_i \in \{0, 1\}$ or $x_i = \{B, F\}$ where $\overset{x_i}{0} \to \overset{d_i}{\text{white}}$, $\overset{x_i}{1} \to \overset{d_i}{\text{red}}$
2. Prior term : $f(x_i, x_j) = |x_i - x_j|$: We encouraged them to be the same.
3. Data term : The idea is that if d_i is white, x_i should be 0 or B, and if d_i is red, x_i should be 1 or F.

   ① Transform x_i's value to d_i's domain → Because $x_i = \{0, 1\}$ and $d_i = \{0, \ldots 255\}$

$$T(x_i) = \begin{cases} \text{white} = (255, 255, 255) & \text{if } x_i = 0 \\ \text{red} = (255, 0, 0) & \text{if } x_i = 1 \end{cases}$$

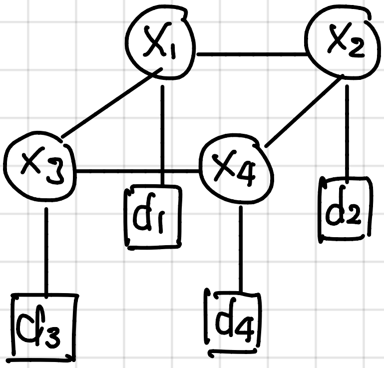
   ② Define a distance function between $T(x_i)$ and d_i.

   e.g.: $f_d(x_i, d_i) = f_d(T(x_i), d_i) = |T(x_i) - d_i|$

Since $x_i = \{0, 1\}$, thus there are 2 possible values of f_d:

$f_d(T(x_i = 0), d_i)$ and $f_d(T(x_i = 1), d_i)$.

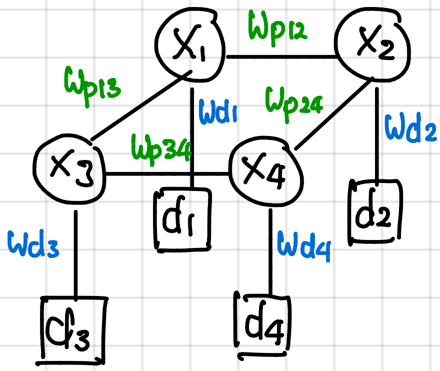
[13] MRFs : Graph Optimization



$$\{x\}^* = \arg\min \sum_i \underbrace{f_d(x_i, d_i)}_{\text{Data term}} + \sum_{j \in N_i} \underbrace{f_p(x_i, x_j)}_{\text{Prior term}}$$

$$\Downarrow$$

Having the values of f_d and f_p for every x_i means that we have a complete weighted graph.



$$\Downarrow$$

Q: How from this weighted graph, can we estimate the optimum values of $\{x\}$?

A: Graph optimization methods

Belief Propagation (BP) | Graphcuts