~~mid lab~~

Final

31/12/2017

Chapter-

Redundancy

same জিনিস বার বার ~~আসা~~। exp: tech fest এ রবি- group কে- কোন topic এর উপর বলতে দিলে more or less 30% কথা আসবে same ধরনের।

~~Redunt~~ compression

কোন কিছুর size কমিয়ে ফেলা।

2 types — 1| lossless compression algorithm (কোন information loss হয় না।)

2| lossy compression algorithm

কিছু information loss হয়। ~~অনেক~~

~~অনেক কমাই~~

exp: ~~আমার~~ কাছে দুটি diff color আছে। যদি 1bit use করি তবে সেটা lossless & যদি same color হবে হবে তখন যদি 4bit use করি, সেটা হবে দুটির জন্য 4bit lossy.

compression Application

1| transmission এর compress করা।

2|

Compression কিভাবে হয়?

Coding Redundancy

R	255, 0, 0
G	0, 255, 0
B	0, 0, 255
Y	255, 255, 0

suppose, কোনো একটি ছবি image আছে। যেটি color represent করতে 2^{24} রকম লাগে। যদি 9 টা pixel থাকে তবে 9×2^{24} bit লাগবে। এখন ধরি, একটি বাংলাদেশের

~~Suppo~~

যদি bangladesh এর flag কে represent করতে চাই, তবে 9×1 এর codebook টা হবে দরকার।

R	0
G	0

=> এই codebook / haffman coding.

9×1 → হলো variable coding

9×2^{24} → হলো fixed length coding.

Interpixel Redundancy

একটি pixel এর সাথে match করে না। একটি pixel, অন্য আরেক pixel এর সাথে describe করে না, তার আশেপাশের এর কোনো pixel অনুযায়ী sequence of pixel যে, একটার pixel হয় কোনো sequence of pixel এর represent করাতে try করে। sequence size কমানো সম্ভব হবে।

Psychovisual redundancy

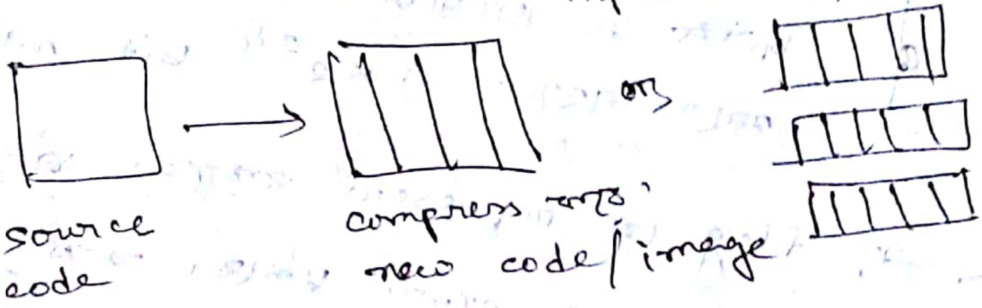
মানুষের brain এর সাথে related. brain এর procedure information কে read করা।

Image compression models

huffman code এর উপর coding করে redundancy কমানো।

~~Source code কে compr~~ $\boxed{\text{fig 6.5 দেখো}}$

input $f(x,y)$, output $\hat{f}(x,y)$



lossless compression

~~কোনো~~ কোনো info miss হয় না।

exp: medical imaging এর ক্ষেত্রে খুবই গুরুত্বপূর্ণ

because কোনো info miss হলে problem হতে পারে

→ Specific example হলো Huffman code কোনো একটি

এর color কে ফিক্স করে আলাদা ভাবে represent করা।

color/code দ্বারা probability এর মাধ্যমে নির্ণয়

⇒ image থেকে color এর

1	7	7	3	9
4	4	2	2	1
1	7	3	3	1
1	7	4	4	1

$P(i) = \frac{6}{20}$ → 1 6 বার আছে ∴ 1 এর probability

20 → total

$P(1)$ কে probability density function বলে।

~~lowest দুটি probability~~

S	P(S)	step 1	step 2	step 3	step 4	step 5	step 6
S_1	0.4	0.4	0.4	0.4	0.4	0.4	0.6
S_2	0.3	0.3	0.3	0.3	0.3	0.6	0.4
S_3	0.1	0.1	0.2	0.2	0.3		
S_4	0.1	0.1	0.2	0.1			
S_5	0.06	0.1	0.1				
S_6	0.04						

এই জোড় যেকোন সাজাবো নিবো

lowest দুটি probability যোগ করব ~~করবো~~

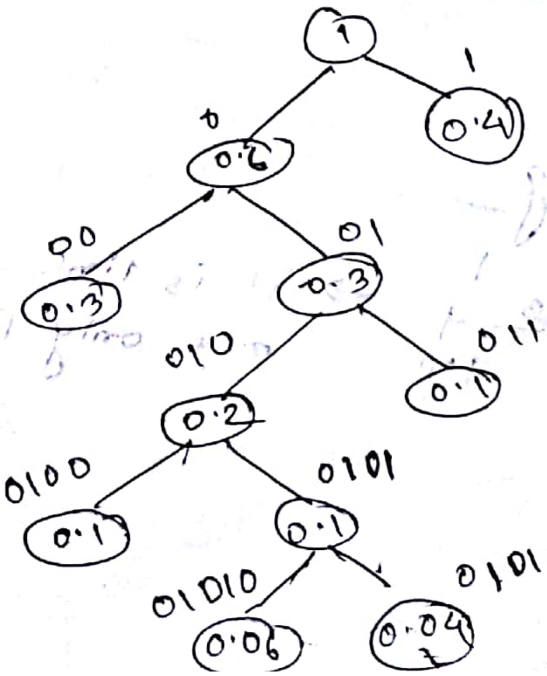
S	P(S)	step 1	step 2	step 3	step 4	step 5	step 6
S_1(1)	0.4	0.4	0.4	0.4	0.4	0.4	0.6 (0)
S_2(00)	0.3	0.3	0.3	0.3	0.3 (00)	0.6	0.4 (1)
S_3(011)	0.1	0.1	0.1	0.2 (010)	0.3 (01)		
S_4(0100)	0.1	0.1	0.2 (0100)	0.1 (011)			
S_5(01010)	0.06	0.1 (0101)					
S_6(01011)	0.04						

01010, 01011

(*) lowest দুটি probability যোগ করবো, P(s) এর 1

যেকোন order এ সাজাবো।

(*) step 3 এর মত যেকোন



So,

S	variable length	fixed length
$s_1(0.4)$	1	000
$s_2(0.3)$	00	001
$s_3(0.1)$	011	010
$s_4(0.1)$	0100	011
$s_5(0.06)$	01010	100
$s_6(0.04)$	01011	101

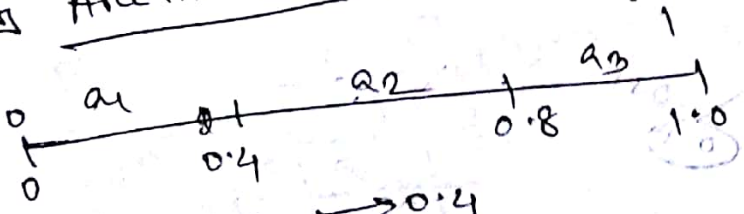
$\rightarrow 0.4\times1 + 0.3\times2 + 0.1\times3 + 0.1\times4 + 0.06\times5 + 0.04\times6$

$= 0.4 + 0.6 + 0.3 + 0.4 + 0.30 + 0.24$

$= \cancel{0.224}\ 2.24$

→ normally মনে হয় যে fixed length টা বেশি better, but actually variable length টা better, because FL এ 3 bit & VL এ 5 bit. as, হিসাব calculation করে 2.24 আসে। as, $2.24 < 3$, so VL is better.

Arithmathic coding –

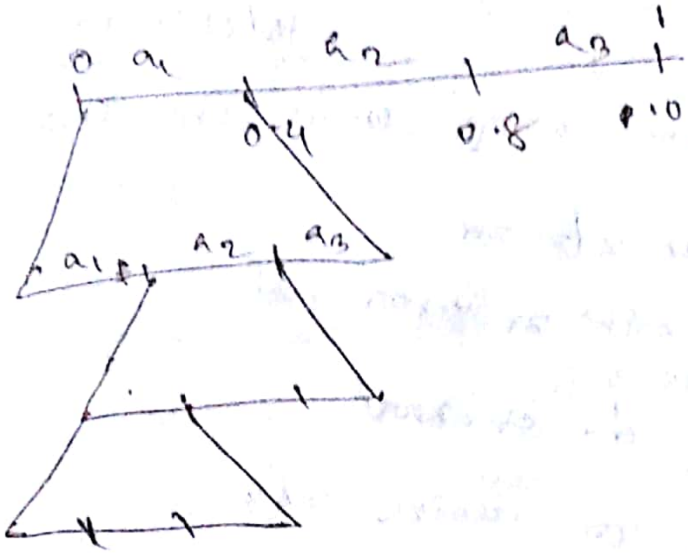


→ it is line segment, not only line.

suppose $a_1 \rightarrow 0.4$

$a_2 \rightarrow 0.4$

$a_3 \rightarrow 0.2$



lab

1) Given image file : ~~image~~

0	1	1	0
0	0	1	1
3	1	1	4

→ # Implement huffman coding.

10/01/2018

- lossless compression → Info same থাকবে, তাই info loss হবে না
- lossy compression → image size কমে যায়।
  accuracy থাকে না বা কোনো কোনো info loss হয়

640 × 840 ⇒ resolution | size of standard

- preprocessor → একটি image কে resize করে।
- compression এর procedure হয় redundancy remove করা।

1	5	9
17	3	21
4	9	91

- coding redundancy : 3টি procedure
  1. coding redundancy
  2. interpixel "
  3. psychovisual "

- coding redundancy :→ 8 bit, 4 bit এ represent করলে no of bit কমে যায়। → একটি pixel vale change করে।

- Interpixel redundancy :→ একটি pixel তার neighbour এর সাথে যে interconnect থাকে।

255	250	251
250	255	250
251	255	251

→ multiple pixel কে একটি pixel দিয়ে represent করা।

Psychovisual redundancy:

→ Visual যে procedure অনুসারে human use করি, তাকে remove করা psychovisual redundancy বলে।

→ ~~lossy~~

→ lossy.

→ আমরা যখন video দেখি তখন কিছু info loss হয়.

→ আমরা same absense mind এ, but তখন যদি আমরা কিছু দেখি না, video টি দেখি তখন আবার নতুন কিছু দেখা যায়, যা 1st বার miss হয়েছিল।

Image ~~compre~~ processing model

1	2	3
4	5	6
7	8	9

1	2	3

→ ~~আমি~~ যে channel এর মাধ্যমে পাঠাবো তার encoding ~~আমি~~ original image পাব।

→ 1st এ decode করা হয়, then

→ $f(x)$ (input) dependent on $f'(x)$ (output) lossy | lossless compression.

$f'(x) = f(x) \rightarrow$ lossless

$f'(x) \neq f(x) \Rightarrow$ lossy

Huffman coding–

→ একটি image এ একটি specific pixel এর probability বের করা

→ 1 আছে 8 বার

1	1	1	1
2	1	1	2
3	1	1	2

$p(1) = \frac{8}{12}$ → total pixel 12টি

↓

আবার 1 হয়েছে probability কত হবে?

Scanned by CamScanner

Same as, $P(2) = \frac{3}{12}$

$P(3) = \frac{1}{12}$

So, $P(1) + P(2) + P(3) = 1$

highest probability এর lowest no দিয়ে represent করে।

Arithmatic coding

→ একটি pixel একবার আলাদাভাবে pixel এর মান নির্ধারণ করা হয়।

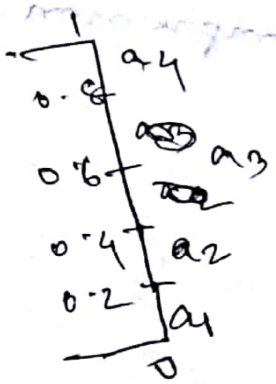
→ a_1 a_2 a_3

a_3 a_3 a_4

→ সবগুলো 1টি number দ্বারা আমরা আমাদের image কে represent করব, একেই বলে arithmatic coding.

→ একটি line segment নিব & 0 থেকে 1 এক করব।

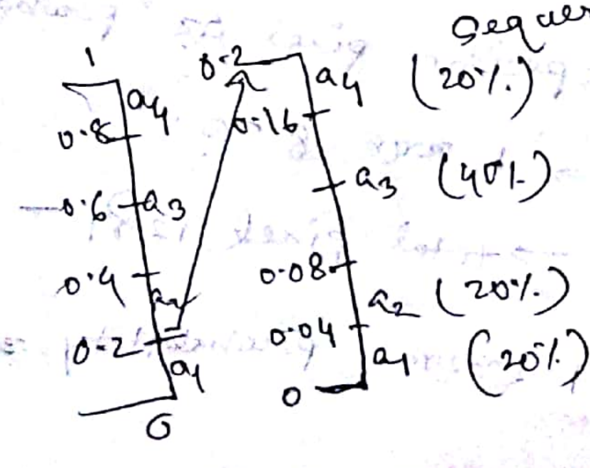
1st step



Symbol	Probability	Interval
a_1	0.2	(0-0.2)
a_2	0.2	(0.2-0.4)
a_3	0.4	(0.4-0.8)
a_4	0.2	(0.8-1)

Sequence: $a_1 a_2 a_3 a_3 a_4$

2nd step



⇒ আমরা, 1st এ interval এ ভাগ করছি। যদি আরও বড় $a_1, a_2 \dots$ থাকে, তাদের individual value কে আবারও interval এ ভাগ।

<u>Formula</u>: (High − low) * probability + New low

<u>Encoding for</u>, $a_4 \to (0.2-0) * 0.2 + 0 = 0.04$

$\to (0.2-0) * 0.2 + 0.04 = 0.08$

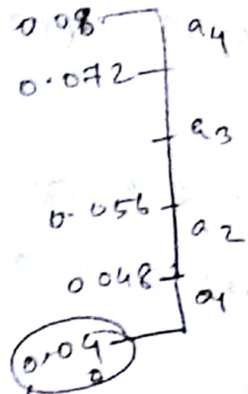
$\to (0.2-0) * 0.4 + 0.08 = 0.16$

$\to (0.2-0)$

$a_2 \Rightarrow (0.08-0.04) * 0.2 + 0.04 = 0.048$

$\to (0.08-0.04) * 0.2 + 0.048 = 0.056$.

$\to (0.08-0.04) * 0.4 + 0.056 = 0.072$.



ছবি sheet/slide এ দেখ

ছবি sheet/slide এ দেখ

<u>Decoding</u>

Suppose, 0.0688 ....

Number − initial value / Probability

$0.068 \to [0 - 0.2] \to a_1$

$\to [0.2 - 0.4] \to a_2$

$(0.068 - 0)/0.2 = 0.34 \to [0.4 - 0.8] \to a_3$.

$(0.34 - 0.2)/0.2 = 0.7 \to [0.4 - 0.8] \to a_3$

$(0.7 - 0.4)/0.4 = 0.75 \to [0.8 - 0.1] \to a_4$

$(0.75 - 0.8)/0.2 = 0.875$

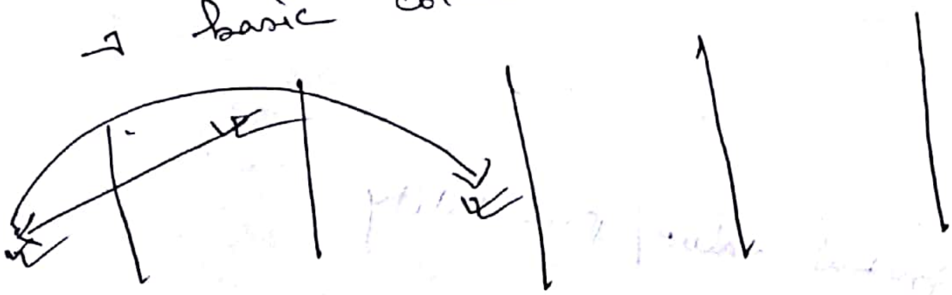
Lempel–Ziv–Welch (LZW)

encoding

0	BLACK
128	GRAY
255	WHITE
256	39-39

39	39	126	126
39	39	126	126
39	39	126	126
39	39	126	126

* pixel being processed এ বর্তমান color serially দিয়ে দেখব read করা হবে।
* 39 এবং প্রথম color read so encoding output এ 39 হবে কেননা 1-255 এর মধ্যে because 1-255
* dictionary location এ 256 হবে already আছে।
→ basic color গুলো



14/01/2018

Fidelity criteria

আমরা যখন image থেকে যখন compression করবো।

→ 2টা image এর মধ্যে difference করে বের করতে হবে।

→ difference, substract করে বের করে।

I		
2	7	9
4	1	3
3	9	2

→

I'		
4	7	9
4	2	3
3	10	2

→ প্রতিটা pixel substract করলে যে যোগ হবে & অনেক error বের হবে। mean-square-error

① সবগুলো যোগফল হবে $\sqrt{\Sigma}$ দিলে mean-square-error বের হবে। → squ mean-square-error

② noise হবে যখন input value ও আউটপুট output value

যে accuracy হবে কম হবে → SNR process / signal to noise ratio

mean-square-error

→ input image & output image এর মধ্যে কতটুকু error fidelity criteria থেকে তা বের করতে প্রতিবারে হবে

④ 5, 7, 7, 7, 7, 7, 179.

$$\text{mean} = \frac{5+7+7+7+7+7+179}{7} = \frac{219}{7} = 31.29$$

median = 7 [middle one]

④ Average থেকে mean square কম হবে।

LZW (Encoding)

→ specific ~~text কে একটি~~ value / text দ্বারা অনেক large area কে represent করতে পারি,

→ প্রতিটি value কে 8 bit দ্বারা represent করি।

→ যখন রঙিন color তখন এটা 3টা channel থাকে,
red, green, blue

→ intensity হলো Gray scale

→

0	BLACK
⋮	
128	GRAY
⋮	
255	WHITE
256	

128 GRAY → suppose

256 → নিজে customizely একটি lookup table এ value input দিচ্ছি

Procedure

1 → যে pixel নিয়ে currently আছি - করতেছি সে সেগুলো pixel being processed এ করিয়ে দিই by serial

→ corresponding output → encoding output

→ এটা সেখানে index এ থাকবে → dictionary location

→ যে সকল pattern describe করছি → Dictionary entry

→ একটি যে একটি establish করেছি →

→ যখন একটি সেগুলো seqⁿ কে একটি currently recognized sequence

X 39 X X X → 1st step এ কোনো output হবেনা।

39 39 39 256 39-39

↓

255 পর্যন্ত color 8 bit এ আসে, তাই 256 হবে।

39 126 39 256 39-126.

126 126 126 258 126-126.

126 39 126 259.

39 39

39-39 126 256..

39-39 specific color গুলোর generate করেছি

→ 256 এ 39-39 ছিল তাই দিলাম

→ Advantage

color গুলো আলাদাভাবে রাখার প্রয়োজন নেই।

LZW (Decoding)

Ques: 39 39 126 126 256 257 258 260 259 257 126.

→

39 → 39

39 39 → 126

126 126 → ...

⋮

~~Prev.~~

two Basic types of image compression

1) ~~ex~~ lossy compression

→ transform coding (FET, DFT, DCT etc) এর জন্য যেকোনো compression এর জন্য এই ৩টা method use হয়।

→ Block truncation compression: অবশ্যই একটি value থাকে, তারপর thresholding হয়।

→ vector quantization compress → ছোট ছোট ভাগে ভাগ করে then process করে।

→ lossy predictive coding

→ wavelet coding

2) loss less compression

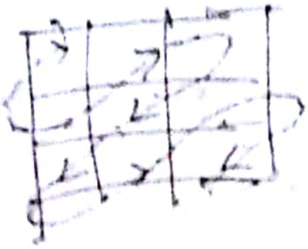
~~los~~ loss compression, lossless compression এ use হয়।

→ Huffman coding

→ Runlength coding: এক্ষেত্রে zigzag coding এর মাধ্যমে use করা হয়।
কোন image এর maximum value zero থাকলে zero কতটি আছে তা represent করা হয়।

zigzag coding

16	0	0	1
0	0	3	0
9	0	0	0
0	4	0	0



16 0 0 0 0 9

⇒ 0, 16   4, 9 → 4 means 9 এর আগে 4 টা zero আছে।
↓
16 এর আগে 0 টি তাই 1st এ zero নাই

→ Block coding:

→ Quad Tree coding: outline represent করে

→

▣ JPEG codec (Joint Photographic Experts Group)
→ অনেক expert রা মিলে এটা তৈরি করেছে।
→ এটি lossy compression.
→ YCBCR: মানুষের চোখের / mind এর সাথে related
↳ brightness → prominence red
↳ blue

Scanned by CamScanner

→ FDCT: (x,y) এর info গুলো DCT করে frequency domain এ নিয়ে যায়।

→ Quantizer:

1। color space transformation: Y CB & CR
Y CBCR

2। Downsampling: 50 × 50 কে array তে convert করব।

3। Block: Block এ ভাগ করব।

4। frequency domain transformation: DCT use করা হয়

5। Quantization: কিছু portion কে easily ফেলতে পারি
But যেই position দরকারী সেগুলো রাখবো।

→ 8×8 image।

6। Encoding: zigzag & huffman coding use করব।

→ Details see in wikipedia GPEG page

Lab

sequence: $a_1 a_2 a_3 a_3 a_4$

symbol	Probability	interval
a_1	0.2	[0 – 0.2]
a_2	0.2	[0.2 – 0.4]
a_3	0.4	[0.4 – 0.8]
a_4	0.2	[0.8 – 1]

→ Arithmatic coding implement [Encoding & Decoding]

24/01/2018

Digital Image Processing

Morphology

Some animal অথবা plants এর feature represent করে

biology

mathematical morphology

~~set theory~~

After segmentation, তারপর morphological operation টি করব।

যা করা হবে set theory র উপর

1	0	1
0	1	0
0	0	0

$FG = \{(1,1),(1,3),(2,2)\}$ represent করতে পারি

binary image : আমরা set এর মাধ্যমে represent

$A = \{1,2,4,3\}$ ✗

$A = \{1,2,3,4\}$ ✓

set এর গুরুত্বপূর্ণ important property হল sorted হতে হবে।

$Z^2 \rightarrow$ দুইটি element এর হয় / 2D এর হয়।

because co-ordinate (x, y) দুটি।

$Z^3 \rightarrow$ 3D এর হয়

$Z \rightarrow$ একটি element / 1D এর হয়।

Basic concepts in set theory

slide দেখো

Morphological operation

1, 0, Black white ?

normally morphological ~~operation~~ operation এ 1 কে consider করি।

↓

noise remove করে

Structuring Elements, Hits & Fits with example.

Fit: সবগুলো match হলে সেটা fit.

যেমন: সবগুলো direct সমান উপর আছে

Hit: কোনো point match হলে সেটা hit

যেমন: সবগুলো সমান ভাবে দিয়ে আছে, গুরুত্বপূর্ণ

সমান উপর আছে না, ডানদিকে আছে।

Miss: কোনো Hit হয়নি কোনো হচ্ছে কোনো হচ্ছে

যেমন: সবগুলো কোনো কোনো হচ্ছে

~~Structing~~ Fitting & Hitting

slide দেখো

Fundamental operation ~~# Eros~~

(1) **Erosion:** only consider fit represent করা হয় $f \ominus s$

f ↓ image, s ↓ structure element

মনে হয়: erosion of image f by structuring element s.

$$g(x,y) = \begin{cases} 1 & \text{if } s \text{ fits } f \\ 0 & \text{otherwise} \end{cases}$$

Scanned by CamScanner

example : অনুরূপভাবে 'fit হলে 1 হবে,
otherwise (hit & miss) হলে 0 হবে।

0 padding use করব।
অনুরূপভাবে 1 consider করব means 1 এর ভিতরে 1 এর মধ্যে 1 হলে, otherwise 0 হবে।
0 কে consider করার প্রয়োজন নেই।
0 কে consider করার প্রয়োজন নেই, ছোট করে ফেলে,

(*) ~~erosion~~ erosion, object কে ছোট করে reduce করে ফেলে
means একটি image এর noise কমায়

formula : erosion for sets A & B in z^2 is defined by the eqn.
$$\boxed{A \ominus B = \{ z \mid (B)_z \subseteq A \}}$$ ⇒ must ভিতরে থাকতে হবে
→ only consider for true, that means 1

$\boxed{\text{book : } 9.2\text{-}3}$

(*) 9.2.2 - Example 1

(2) Dilation ~~যেকোনো একটা~~ এবার যাবে। 1 হয় means orientation এ
(*) dilation এ ~~কোনো~~ only consider hit

$f \oplus s$

$$g(x,y) = \begin{cases} 1, & \text{if } s \text{ hits } f \\ 0 & \text{else} \end{cases}$$

only 1 এর ~~এর~~ জন্য hit consider করব।
(*) ~~hit হবে~~ ~~only 0~~ means একটি 1 এর সাথে 1 match করবে, 0 match করলেও
বাদ করলে কোনো সমস্যা হবে ~~নেই~~ নেই।

* dilation, image এর size বড় করে।
hole fill করার জন্য use করা হয়।

* formula: Dilation of A by B & is defined by eqn

$$A \oplus B = \{ z \mid (\hat{B})_z \cap A \neq \emptyset \}.$$

* ~~Dilation : Example~~

* 9.2.1 Dilation: Example 1 & 2

Compound Operations

① Opening: erosion followed by dilation

$$f \circ s = (f \ominus s) \oplus s.$$

1st এ erosion করে যেটি পাবে & তারপর এর উপর dilation করে output image কে বড় করবে।

erosion করে match check করবে।
then dilation করবে, dilation করার সময় 1st এর object এর সমান হবে, new object তৈরি হবে না।

② Closing: ~~উল্টা~~ opening এর উল্টা।

$$f \bullet s = (f \oplus s) \ominus s.$$

hole remove করার জন্য use করা হয়।

9.4: The Hit-or-Miss Transformation

যদি hit হয় তাহলে 1

miss " " 0.

Hit-or-miss Transformation : Example

$\cancel{A\oplus}$. $A \odot x = (A \ominus x) \cap (A^c \ominus (w - x))$

* w, structure image এর বাকি or হয় হবে
& 1, এর red color দেখা। ও এর orientation

* slide এর example
* book এর example : next cls.

~~31/02~~ 28/01/2018

31/01/2018

9.4: The Hit or Miss Transformation

Formula

$$A \circledast X = (A \ominus X) \cap (A^c \ominus (W - X))$$

→ 0 padding করতে হবে, A তে।

→ A^c হল A এর complement

→ $A \ominus X$ করতে হবে. 1 1 → only 1 থাকলে তবে এটা নিবো। otherwise 0

→ $A^c \ominus (W-X)$ " " 1 1 " " 1 " 0

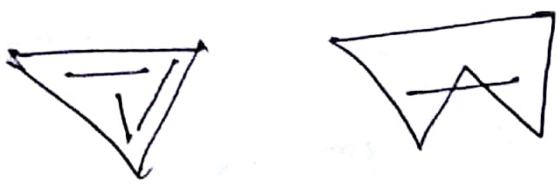
↳ but এটা এর জন্য আগের 1 or 0 নিতে হবে।

→ example: Digital লাইব্রেরি থেকে কোন character search করা। আমরা যে transform করবো সেটা খুঁজে যেখানে পাবো সেখানে 1 দিব।

9.5.4: Convex Hull

Formula

$$X_k^i = (X_{k-1} \circledast B^i) \cup A.$$



একটি object কে represent করতে চাইলে সব জায়গা represent করা যায়। সব বাইরে point দ্বারা represent করা যায়।

(*) convex Hull থেকে original object কে বাদ দিলে যা থাকে সেটা হল convex deficiency

→ shaded area ~~represent~~ represent করছে 1

white " " " 0

x " " " 0 or 1 হলেও কোন problem হবে না।

→ 4 টি point (structure নিয়ে)

→ Given structure element থেকে, shadow অংশ ঠিক রেখে বাকিগুলো কে 1, 2 - - ঘুরাতে হবে। (ঘুরিয়ে করা যায়)

→ Problem: finally যে image আসে তা দিয়ে সম্পূর্ণ image কে ~~~~ represent করা যাচ্ছে।

→ এখানে নির্দিষ্ট boundary set করবে। (4 টি point নিয়ে) এই boundary র বাইরে কোন point allow করবে না।

9.5.5 thinning

(*) Thin means ~~অর্থাৎ~~ object কে ছোট করা

(*) formula

$$A \otimes B = A - (A \circledast B) = A \cap (A \circledast B)^c$$

$\otimes$ → symbol

$(A \circledast B)$ → structure set

→ 8 টি ~~~~ structure নিব।

→ Hit or miss transform করব 1st structure element বা উপরে রেখে, উল্টে।

(0 padding করব)

→ original image থেকে বাদ দিব

→ এই সময় যেই image দিব তার 2nd structure element

~~এ-র~~ অনুযায়ী hit or miss transform হবে।

~~Thickening~~

9.5.6: Thickening

⊛ Formula

$A \odot B$

→ যেটির structure element এর উপর hit or miss transform হবে,

→ then original এর সাথে new টি union হবে।

9.5.7 : Skeleton Example

Basic shape কে represent করার জন্য যেসব point দরকার হয়

⊛ Formula

$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

with opening point দিয়ে যেসব জায়গায় more করা যায় only ঐ point গুলো নিব।

Pruning

only defn জানতে হবে।

exp লাগবে না।

Scanned by CamScanner

Segment - 6

Image Segmentation

না পড়ালেও হবে
শুধু নাম only এই page টা।

Introduction

- Overview of segmentation Techniques
- Edge-based techniques only edge identify করতে পারে
- Region-filling
  identify করতে পারে
  সম্পূর্ণ region
- ~~ST~~ color
- Data clustering via kmeans
- Texture-based techniques ... করতে হয়, তার maximum
  image processing এর কাজ করতে
  texture based -
  specific texture কে (suppose, motion) detect করতে পারে
  তাই হল texture based techniques
- Motion segmentation ... এর কোন ~~object~~
  প্রথমে background image থেকে ... কোন object বের হয়, যেখানে
  image দিয়ে দিলে original
  হল motion segmentation
- Document segmentation
- Biometric Image segmentation

Segmentation : Preview

(*) dis continue এর ~~exple~~ example হল thresholding

Point Detection

example :- (details from slide)

~~Suppos~~ line masks :

$$f(x,y) - f(x,y-1) + f(x,y) - f(x,y+1)$$

Line Detection

line কে detect করা হয়েছে (fig 10.4)

Edge Detection

(*) line হল মোটামুটি info বোঝায়
but edge হল abrupt change হয়েছে (যেমন,  এখানে ~~edge~~ এক side এ black but অন্য side এ white যেমন edge এর color change হয়েছে, এটাই হল abrupt change.

(*) line এ নির্দিষ্ট value টা important
but edge এ পার্থক্যের value এর important.

Gradient Operators

দুটি specific point এর মধ্যে difference টা যেখানে এক জায়গা যেমন কোনখানে color অনেকটা বেশি change হয়েছে।

$G_y = f(x-1, y) - f(x+1, y)$

mag(∇f)

$\nabla f = \left[G_x^2 + G_y^2\right]^{1/2}$ ⇒ কতটুকু change হয়েছে

$\alpha(x, y) = \tan^{-1} \dfrac{G_x}{G_y}$ ⇒ কোন দিকে change হয়েছে।

Q) Edge linking & Boundary Detection

→ information missing হলে

→ ভাঙ্গাভাঙ্গি edge এর (— —) এর magnitude same হলে & দিক ও একই হয় তাহলে তাদের মধ্যে link করে দিব

→ magnitude কম না হলে তাকে omit করি।

Q) Thresolding

→ value কোন লেভেল কম হলে বাদে বলে Thresolding

→ 2 type of thresolding

① Global Processing : সম্পূর্ণ image এর উপর same info & procedure use করবো।

suppose : 0 – 20.

② Local thresolding : প্রতিটা mask এর highest & lowest কোন portion এর উপর point নিয়ে & তাদের প্রত্যেকটা thresolding apply.

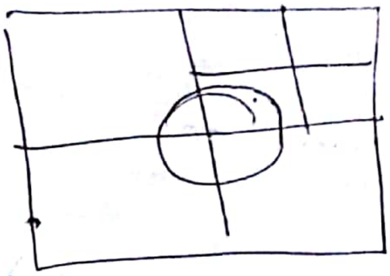
adapting threshholding : .. প্রয়োজন অনুযায়ী threshold value টা adapt করবো।

Region Growing –

যে কোন point থেকে region অনুযায়ী অনুযায়ী বড় হতে থাকে। same color হলে region বড় হবে।

Region split: full image কে চারটি image দিয়ে।

~~then~~,



then region এর ভিতর- ভিতরে same color দিলে, যদি না হয় তবে 4টা ভাগ করা হবে। again ওই full region এর ভিতর 1টা part দিয়ে যদি তারপর same color না হলে 4টা ভাগ করা হয়।

03/02/2018

Detection of Discontinuities

Detection of Discontinuities line Detection.

Detection of Discontinuities Edge Detection

$f(x+1, y+1)$ & $f(x-1, y-1)$ থেকে $f(x, y)$ থেকে ~~...~~ দূরত্ব বের করা হয়।

wikipedia তে sobel operator দিয়ে search

Edge linking & Boundary Detection local processing

Thresholding

□ → $\boxed{\text{Otsu method}}$:

□ → Adaptive Thresholding / Local thresholding — ex: suppose Engineering dept. এর গড় 3.5 হলো। আবার waiver হলো কিন্তু EEE এর গড় 3.0 হলো। তাহলেও image / object ভেদে value differ করে।

→ global thresholding — সবার এক same value / range সবগুলো image এ কে fix করা।

□ color-based segmentation

□ region-based (similarity based) segmentation

→ একটা region কে fill করতে।

→ 5 টা codition আছে (slide এ দেওয়া)

□ region-based segmentation : Basic formula

□ Region growing

region এর থেকে শুরু সম্পূর্ণ region কে নিয়ে নেয়।

□ Region splitting

region গুলো same কিনা দেখে যেখানে 4 টা করে portion এ ভাগ করা।

□ Use of motion in segmentation

→ ~~ADI~~ বা video তে মুভমেন্ট কাজ করে।

see in book

→ character কে কিভাবে segmentation করে?

→ English sentence segmentation কিভাবে করে?

lab

Segmentation

```
input = imread(' ');

SE = strel('square', 7);

Output = imdilate(input, SE);
```

→ ~~Stru~~.

bwhitmiss

bwmorph

imclose

imopen

imerode

imdilate

→ (input, 'skel', Inf)

→ dilate করে erode

↓ বড় হয়ে যায়

↓ ছোট হয়ে যা

```
input = imread('circles.png');

SE = strel('square', 3);

Output = imdilate(input, SE);
```

$BW = input - (input \ominus SE)$

Color-based segmentation

input RGB image দিব

output এ binary image আসবে।

→ printout copy জমা দিতে ~~হবে~~ হবে।

Assignment Name: face region segmentation

code

Output:

input image	output image

11/02/2018

Seg-5 & 6 => CT
18/02/2018

Chapter-11 (segment-7)

Presentation & Description

1) chain code
2) Polygon base method
3) Boundary base method
4)

Chain codes

easiest way একটি code কে represent করে।

Fig 11.2

Polygonal Approximations

1) Minimum parameter polygon

ছোট ছোট ঘরে ভাগ করে। then, সবে থেকে distance কম
যেগুলোর থেকে ছোট ওইটুকু portion নিয়ে নিব।

Fig 11.3

Splitting techniques যেকোন 2টি point নিব & যেখান থেকে line
সবচেয়ে দূরবর্তী ... threshold থেকে বেশি হয়
consider করব, যদি এটি ... point কে ভাগ করব।
~~তখন~~ আবারও ছোট ছোট point ... but এটি
→ Splitting এ বেশি error generate হবে but এটি faster ~~calcu~~ কাজ করবে।

☒ Signature

~~Image / object কে 1-D তে convert করা।~~

object এর center position থেকে সব দিকে আলো ফেলে একটি new shape (any type) generate হবে
দিয়ে আলো ফেলবে একটি new shape (any type) generate হবে

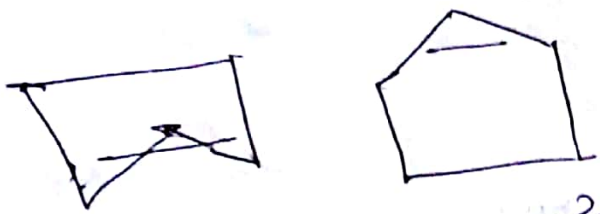
তারপর 2-D image কে 1-D তে convert করা, angle & distance এর মাধ্যমে

☒ Boundary segments

~~Convex Hull:~~

bounding box হল boundary segments বের করে
বের করার ~~উপায়~~ পদ্ধতি।

(*) Convex Hull: one of the easiest way of boundary segments



convex dificency: ভেতরের extra portion গুলোকে বোঝায়।

object & convex hull যেখানে meet করবে সেগুলোকে ~~ছবি~~ তারপর করে mark করে।

☒ Skeleton

যে object থাকে তাকে graph এ represent করব,
graph representation

advantage: সবসময় graph representation

1967 এ proposed করা হয় এই method.

multiple boundary ~~space~~ pixel যে point থেকে সমান তা
হল skeleton & এর output হয় all time graph.

~~eha~~ ⊞ Shape number

(+) সুবিধা : object same number হবে

chain code :

difference : 1st & last এর difference বের করে & shape no হয়। এ ১ম পরে (?) বের করে।

shape no : chain code এর যেটা ছোট সেটা ~~first~~ difference হবে & clockwise count করবো।

(+) shape no হলো যখন conjugative value হ মানে - তাই এর ~~আগের~~ shift করেছি।

⊞ Topological Description

characteristics এর shape ~~lists~~ difference একটি object এর change হয় না।

এর কারণে কোনটির change হয় না।

Eucledian number (E) = Connected component (C) - Hole (H)

Eucledian number = component - Hole

vertex - edge + face = Eucledian number

or $V - Q + F = E = C - H$

~~$\Rightarrow$ 11 - 7 + 3 =~~

$\Rightarrow 7 - 11 + 3 = 1 - 3$ ...

$\Rightarrow -1 = -2$

confusion see book

⊞ Texture.

$\boxed{\text{xm এ আসবে না}}$

⊞ statistical method

$\boxed{\text{not important for xm}}$

* Slide 32 to 38 => ans

* slide 45 to 61 => ans

* string representation & tree base method
  fig 11·34 & 11·35

---

lab

1) Sobel operator
2) ~~Prewitt~~
3)

Solved

Img

	5	7	4	
	9	11	4	
	10	13	9	

Gx

1	0	-1
2	0	-2
1	0	-1

Gy

1	2	1
0	0	0
-1	-2	-1

$$img = \sqrt{Gx^2 + Gy^2}$$

$$Gx = (5 + 2*9 + 10) - (4 + 2*4 + 9)$$

$$Gy = (5 + 2*7 + 4) - (10 + 2*13 + 9)$$

$$output = \sqrt{Gx^2 + Gy^2}$$

$$\theta = \tan^{-1}\left(\frac{Gy}{Gx}\right)$$

Prewitt

1	0	-1
1	0	-1
1	0	-1

1	1	1
0	0	0
-1	-1	-1

APP

BOOFCV

Demonstration

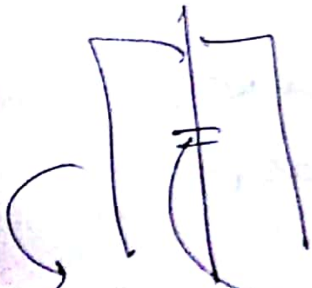
14/02/2018

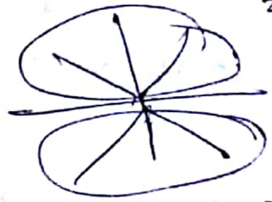
v. v. I

Chain Code এর সাহায্যে একটা specific ... এর area বের করা

Calculate the area, perimeter & shape factor of structure using chain code

top-most left side থেকে area বের করব।

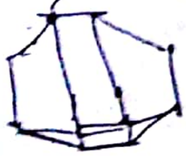




Area = - - -

perimeter = even count + $\sqrt{2}$ odd count

Shape factor = perimeter^2 / Area



Chapter-12 → Object Recognition

(Segment-8) ⇒ (pattern recognition এর part)

* Detection & recognition most of কেসে machine learning

* **Pattern recognition**: এখানে object এর different type of pattern দ্বারা recognize করে।
  — আবার এই feature কে ব্যবহার করে pattern আকারে রাখে

* 1-D array কে vector বলে

* Feature vector → collection of feature, Array দ্বারা represent করা হয়।

→ String Array দ্বারা pattern represent করা হয়।

Application area of P.R

→ pattern recognition হল machine learning.

→ (slide দেখো)

Pattern Recognition System

Pattern & pattern classes

Pattern & pattern classes: Another Vector Example

" " " " : String Example

" " " " : Tree Example

Recognition based on Decision-theoretic Methods

→ সবচেয়ে popular method হল statistical method

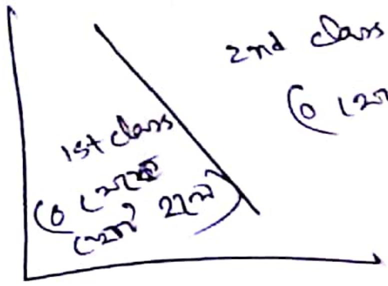
→ $\begin{pmatrix} 4.3 & 1.3 \end{pmatrix}^{T}$ & $\begin{pmatrix} 1.5 & 0.3 \end{pmatrix}^{T}$

T represent করতেছে এটা হল transpose matrix.

$d_1(x) =$ $\qquad\qquad$ $d_2(x) =$

then $d_{12}(x) = d_1(x) - d_2(x)$

$= \cdots\cdots$ → এখানে যেই result আসবে তা ~~0 < হলে~~ 0 থেকে বেশি হলে 1st class & ছোট হলে 2nd class.



▣ Template matching —

→ এটা হল some kind of object detect করা।

→ binary image দ্বারা হবে & match করবে।

▣ Recognition Based statistical method

$\boxed{\text{পড়াবে না}}$

▣ Recognition Based on Decision-Theoretic methods neural network

দিবে & সে detect করবে যেই রকম বলে দিতে

→ 1000 of info দিয়ে

কোনটা better

→ input এ হবে node যেমন: length & pattern.

& output এ হবে আবার: যেমন 1st class & আরেকটা 2nd class

↳ last এ যা আসবে

মাঝখানে: hidden layer

→ output যে আসবে input এ আবার দেয়া হয় backpropagation.

→ all definition & why type এর ques.
এইগুলো দেখে নিতে হবে।

18/02/2018

lab
[for final xm]

1. huffman
2. arithmatic
3. LZW
4. erison
5. dilation
6. opening
7. closing
8. edge detection