

LATVIJAS UNIVERSITĀTE
FIZIKAS UN MATEMĀTIKAS FAKULTĀTE
MATEMĀTIKAS NODAĻA

**NESTRIKTAS UN VARBŪTISKAS METODES ATTĒLU
APSTRĀDEI: TEORĒTISKIE ASPEKTI UN LIETOJUMI ROBOTU
VADĪBAI**

DIPLOMDARBS

Autors: **Jana Ivanova**

Stud. apl. nr. ji06009

Darba vadītājs: prof. Aleksandrs Šostaks

RĪGA 2011

Anotācija

Attēlu apstrādei ir plašas pielietojuma iespējas. Viena no jomām, kur tā ir nepieciešama, ir robotika.

Balstoties uz prakses laikā veiktajiem pētījumiem attēlu segmentācijā robotam IURO, tiek apskatītas trīs attēlu apstrādes metodes - Markova gadījuma lauks, nosacītais gadījuma lauks un nestriktu metriku filtrs.

Mērķis ir iepazīties ar minēto algoritmu darbību, to priekšrocībām un metožu matemātisku pamatojumu. Iegūtās teorētiskās zināšanas izmanto, lai dotos modeļus piemērotu attēlu apstrādei.

Sasniegtie rezultāti ir dziļāka izpratne par attēlu apstrādes algoritmiem no matemātiskā viedokļa un saiknes izveide starp dotajām metodēm un to izmantošanu attēlu apstrādē.

Atslēgas vārdi: nestrikta kopa, nestrikta metrika, Markova gadījuma lauks, nosacītais gadījuma lauks.

Abstract

Image segmentation has numerous applications in various fields, one of them being robotics. Based on research in image segmentation for the IURO robot accomplished during a study internship, three methods of image segmentation will be observed in the following thesis: the Markov random field, the conditional random field and the fuzzy metric filter.

The objective is to present these three algorithms, unveil their advantages and analyze their mathematical justification in order to provide their practical demonstration afterwards.

In the first part, the obtained results give a higher knowledge of algorithms used in image segmentation from the mathematical/theoretical point of view. In the second part, the application of the given methods in image segmentation is demonstrated.

Keywords: fuzzy set, fuzzy metric, Markov random field, conditional random field.

SATURS

IEVADS	6
1. NESTRIKTAS KOPAS.....	8
1.1. Nestriktas kopas definīcija	8
1.2. Vēsturiskā attīstība	9
1.3. Pielietojuma piemēri	10
1.4. Nestriktu kopu raksturlielumi	10
1.5. Darbības ar nestriktām kopām	12
1.5.1. Nestriktu kopu šķēlums un apvienojums	13
1.5.1.1. Nestriktu kopu šķēlums.....	13
1.5.1.2. Nestriktu kopu apvienojums	14
1.5.2. Nestriktu kopu papildinājums.....	14
1.6. Nestriktu kopu operāciju īpašības	16
2. NESTRIKTAS METRIKAS UN TO PIELIETOJUMS ATTĒLU FILTRĒŠANĀ..	18
2.1. t-normas jēdziens	18
2.2. Nestriktas metrikas jēdziens.....	19
2.3. Nestriktu metriku piemēri	20
1. piemērs	20
2. piemērs	24
2.4. Attēlu filtrēšana, izmantojot nestriktās metrikas.....	27
3. MARKOVA GADĪJUMA LAUKI UN NOSACĪTIE GADĪJUMA LAUKI	32
3.1. Statistiskā fizika no neizolētām sistēmām	32
3.1.1. Bolcmana sadalījums	32
3.1.2. Sistēmas ar mainīgu daļiņu skaitu	33
3.1.2.1. Gibbsa sadalījums	33
3.2. Apzīmējuma piešķiršana attēlu analīzē.....	34
3.2.1. Mezgli un apzīmējumi	34
3.2.2. Apzīmējuma piešķiršanas problēma	35
3.2.3. Apzīmējuma piešķiršana pie dotiem ierobežojumiem.....	36
3.3. Optimālā atrisinājuma meklēšana attēlu apstrādē.....	37
3.4. Matemātiski MRF modeļi	37
3.4.1. Markova gadījuma lauks un Gibbsa sadalījums	37

3.4.1.1. Kaimiņu sistēma un kliķes	38
3.4.1.2. Markova gadījuma lauks	40
3.4.1.3. Gibsa gadījuma lauks	42
3.4.1.4. Markova-Gibsa ekvivalence.....	43
3.4.2. Enerģijas funkcijas loma.....	45
3.4.3. <i>Ising</i> modelis.....	45
4. NOSACĪTAIS GADĪJUMA LAUKS	47
5. MARKOVA GADĪJUMA LAUKA UN NOSACĪTĀ GADĪJUMA LAUKA IEGŪTIE REZULTĀTI.....	49
NOBEIGUMS.....	50
IZMANTOTĀ LITERATŪRA UN AVOTI.....	51
PIELIKUMI	53
1. boltzman.....	53
2. rbmBB.....	53
3. examplecode.....	56
4. image_labeling	58
5. boltzman_label	63
6. boltzman_backward	63
7. adjacency.....	64
8. CRF_Training	65
9. label_images.....	67
10. feature.....	68
11. quartile	68

IEVADS

Darbs ir izveidots, balstoties uz pieredzi, kura iegūta prakses laikā Minhenes Tehniskās universitātes institūtā. Institūta nodaļa, kurā norisinājās prakse, pilnveido Interactive Urban Robot (IURO, lat. interaktīvais pilsētas robots). IURO projekts paredz radīt robotu, kas patstāvīgi kustas nestrukturētā pilsētas apkārtnē un atrod ceļu, komunicējot ar cilvēkiem. Rezultāta sasniegšanai tiek kombinēti pētījumi no dažādām zinātnes un tehnikas jomām - patstāvīgā kustība, ceļa plānošana, apkārtnes modelēšana un cilvēka - robota ietekme. Autora apskatītā sfēra ir attēlu apstrāde. Robots no iebūvētām kamerām iegūst attēlu, kas ir jāsegmentē, respektīvi, ir jāiemāca IURO, kas ir konkrētais objekts. Robotam ir svarīgi atšķirt šķēršļus un izvairīties no tiem, palikt uz ietves un atpazīt cilvēkus, lai varētu tos uzrunāt un iegūt no tiem informāciju.



IURO robots institūta gaiteni

Mērķis ir iepazīties ar algoritmiem, kurus izmanto attēlu segmentācijai. Ņemot vērā iegūtās praktiskās zināšanas, darbā tiek aplūkoti trīs matemātiski modeļi attēlu apstrādei – nestriktu metriku filtrs, Markova gadījuma lauks un nosacītais gadījuma lauks. Interese par nestriktajām metriķām radās kā turpinājums kursa darbam, kurā tika pētītas nestriktas kopas. Markova gadījuma lauks un nosacītais gadījuma lauks apskatīti prakses laikā, šī iemesla dēļ uzmanība tiek pievērsta abiem modeļiem.

Darbs ir sadalīts trīs daļās. Pirmajā daļā tiek aplūkots nestriktas kopas jēdziens, dots nestriktu kopu raksturlielumu pārskats un aplūkotas darbības ar nestriktām kopām.

Otrā daļā tiek apskatīti nestriktu metriku piemēri, kā arī sniegts pierādījums, ka tās ir nestriktas metriķas. Šie piemēri ir izvēlēti un definēti tā, lai tos varētu izmantot attēlu apstrādē. Paskaidrots attēlu filtrēšanas algoritms, kurā izmanto nestriktas metriķas. No divām apskatītajām nestriktām metriķām izveido vienu unikālu, lai reducētu trokšņus attēlā. Ir parādīti sasniegtie rezultāti, t.i., ar filtru apstrādātie attēli.

Trešā daļā uzmanība ir pievērsta Markova gadījuma laukam un nosacītajam gadījuma laukam. Šīs metodes izmanto attēlu segmentācijā. Vispirms ir dots ieskats fizikālās sistēmās un aplūkoti izmantotie Bolcmana un Gībsa sadalījumi. Turpinājumā tiek izveidota analogija starp attēliem un statistiskām mehāniskām sistēmām. Pikseļa krāsu līmeņi un šķautņu orientācija tiek uzskatīti kā atomu vai molekulu stāvokļi fizikālās sistēmas režģī. Teorētiski ir

aprakstīti Markova gadījuma lauks un nosacītais gadījuma lauks, kā arī definēti nepieciešamie jēdzieni, lai raksturotu šos modeļus. Piemērs attēlu apstrādei, kurā izmanto fizikālu sistēmu, ir *Ising* modelis. Tiek paskaidrota pētījuma problēma, optimālā atrisinājuma meklēšana un parādīti attēli, kuri ir iegūti ar šo algoritmu palīdzību.

Pielikumā var iepazīties ar autora uzrakstītajām un papildinātajām programmām attēlu apstrādei prakses laikā. Visi kodi ir rakstīti Matlab programmā.

1. NESTRIKTAS KOPAS

Nodaļā tiek apskatīts nestrikta kopas jēdziens, atsaucoties uz literatūras sarakstā minētajiem avotiem [10], [11], [12], [13], [15]. Sniegts ieskats nestriktajā kopu teorijā, tās vēsturiskajā attīstībā un tiek aplūkotas tās izmantošanas sfēras. Ir dots nestriktu kopu raksturlielumu pārskats un pētītas darbības ar nestriktām kopām – to šķēlums, apvienojums un papildinājums.

1.1. Nestrikta kopas definīcija

Klasiskā Kantora kopu teorija apgalvo, ka elements atrodas kādā kopā vai ne. Šis izteikums kopai M tiek attēlots sekojoši ar karakteristikisko funkciju:

$$c_M(x) = \begin{cases} 0, & x \notin M \\ 1, & x \in M \end{cases}$$

Nestriktai kopai piederības jēdziens tiek paplašināts līdz intervālam $[0,1]$.

1. definīcija Par universālu kopu X sauc kopu, kas sastāv no visiem pētāmā apgabala elementiem.

2. definīcija Dota universālā kopa X , tad kopu $A = \{(x, m_A(x)) \mid x \in X\}$ sauc par nestriktu kopu, kur $m_A : X \rightarrow [0,1]$ ir piederības funkcija. Tas nozīmē, ka katram x no kopas X tiek piekārtota piederības vērtība starp 0 un 1 (ieskaitot vērtības 0 un 1).

No definīcijas var redzēt, ka nestrikta kopa A tiek raksturota ar piederības funkciju.

[15]

1.2. Vēsturiskā attīstība

Nestriktās loģikas saknes meklējamas antīkajā Grieķijā. Jau filozofs Platons postulēja, ka starp jēdzieniem „paties” un „nepaties” atrodas trešais apgabals. Pirmās nestriktās sistēmas 1920. gadā ieteica J. Lukasevičs. Novērojums: termini kā garš, vecs vai karsts ir grūti attēlojami ar Aristoteļa patiesības jēdzieniem paties vai aplams $\{0,1\}$. 20. gados J. Lukasevičs paplašināja loģikas sistēmu uz visiem reālajiem skaitļiem $[0,1]$. Skaitlis no intervāla $[0,1]$ apraksta iespējamību, ka izteikums ir paties.



Nestrikto kopu teoriju 1965. gadā attīstīja L. A. Zadeh, savulaik elektrotehnikas profesors Kalifornijas Universitātē, ASV. L. A. Zadeh attīstīja jaunu veidu, kā rīkoties ar neskaidru informāciju, lai vienkāršotu kompleksu sistēmu modelēšanu.

1.1. att. L. A. Zadeh

Nestriktā teorija piedzīvoja savu uzplaukumu 1980. gados Japānā ar tā saukto „nestrikto vilni”. Vēsturisks piemērs ir automātiskā metro regulēšana, pirmais lielais veiksmīgais nestriktās loģikas pielietojums praksē, kas ar nestriktās regulēšanas palīdzību padara iespējamu patīkamu piebraukšanu un bremzēšanu. Vēlāk nestriktā loģika atrada plašu pielietojumu izklaides elektronikā. Eiropā „nestriktais vilnis” aizsākās tikai 1990. gadu vidū.

[11], [13]

1.3. Pielietojuma piemēri

Nestriktā loģika mūsdienās tiek izmantota dažādās nozarēs: pielietojums tai ir tādās jomās kā automatizācijas tehnika, medicīnas tehnika, mākslīgais intelekts, valodas atpazīšana, izklaides elektronika, transportlīdzekļu tehnika, regulēšanas tehnika u.c. Šajā gadījumā no valodnieciski izveidotiem teikumiem ar nestriktās loģikas palīdzību tiek iegūts matemātisks apraksts, kas var tikt izmantots skaitļošanas ierīcēs. Piemēri ir videokameras, kas automātiski līdzsvaro nelielas cilvēka trīcēšanas kustības, veļas mazgājamās mašīnas, kas automātiski noregulē mazgāšanas režīmu, ņemot vērā veļas netīrības pakāpi.

Ar nestriktās loģikas palīdzību var izvairīties no paradoksiem, kuri citādāk digitalizācijas procesā radītu problēmas, piemēram, „no cik gadiem cilvēks ir vecs?” vai „no kāda augstuma pakalns ir kalns?”. Šie jautājumi praksē parasti nesagādā problēmas, bet to konkrēta noteikšana rada jocīgas nejēdzības. Nestriktā loģika palīdz novērst šīs dīvainības.

[11]

1.4. Nestriktu kopu raksturlielumi

3. definīcija Nestrikta kopas A nesējs (*angļu val.* support) universālā kopā X ir kopa no visiem X elementiem, kuriem ir pozitīva piederības funkcijas vērtība:

$$\text{supp}(A) = \{x \mid m_A(x) > 0, x \in X\}.$$

4. definīcija Tolerance *toll* apraksta, kurā intervālā $[m_1, m_2]$ piederības funkcijas vērtība ir 1

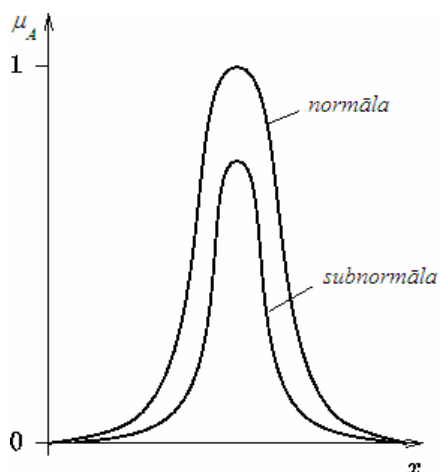
$$\text{toll}(A) = [m_1, m_2] = \{x \mid m_A(x) = 1, x \in X\}.$$

5. definīcija Ja A ir nestrikta kopa X , tad

$$H(A) = \max \{m_A(x) \mid x \in A\}$$

sauc par kopas A augstumu.

6. definīcija A sauc par normālu nestriktu kopu, ja ir spēkā $H(A) = 1$, pretējā gadījumā to sauc par subnormālu (salīdzinājumam skat. 1.2. attēlu).



1.2. att. Normāla un subnormāla nestrikta kopa

[12]

7. definīcija Dots $a \in [0,1]$, X universālā kopa, A nestrikta kopa un $m_A : X \rightarrow [0,1]$ tās piederības funkcija, tad kopu $A_a = \{x \mid m_A(x) \geq a, x \in X\}$ sauc par a līmeņa kopu un $A^a = \{x \mid m_A(x) > a, x \in X\}$ par stingru a līmeņa kopu.

[15]

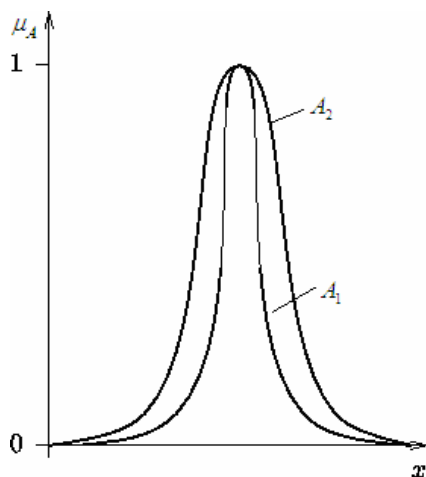
8. definīcija Kopas A 1-līmeņa kopu $A_1 = \{x \mid m_A(x) = 1, x \in X\}$ sauc par A kodolu.

9. definīcija Par galīgas nestrikta kopas A apjomu $C(A)$ sauc $C(A) = \sum_{x \in X} m_A(x)$.

10. definīcija Nestriktu kopu A_1 sauc par nestrikta kopas A_2 apakškopu universālā kopā X , ja

$$m_1(x) \leq m_2(x) \quad \text{visiem } x \in X.$$

Pieraksts: $A_1 \subseteq A_2$.



1.3. att. Nestrikta apakškopa

1.5. Darbības ar nestriktām kopām

Ir vairākas iespējas, kā definēt nestriktu kopu šķēlumu, apvienojumu un papildinājumu. Vispārīgi tiek prasīts, lai nestriktu kopu operācijas būtu savienojamas ar klasiskajām kopu operācijām, tā kā klasiskas kopas izveido nestriktu kopu speciālu gadījumu.

Klasiskās kopu operācijas var tikt definētas ar atbilstošajām karakteristikajām funkcijām. Klasiskās kopas var tikt apstrādātas ar sekojošām aprēķinu metodēm: dotas A un B klasiskas kopas no universālās kopas X un $c_A(x)$, $c_B(x)$ to karakteristikās funkcijas. Tad ir spēkā:

1. $c_{A \cap B} = \min(c_A, c_B)$

2. $c_{A \cup B} = \max(c_A, c_B)$

3. $c_{A^c} = 1 - c_A$

Nestriktu kopu šķēlums, apvienojums un papildinājums tiks definēti tāpat kā klasisku kopu attiecīgās operācijas. Tātad nestriktu kopu šķēluma, apvienojuma un papildinājuma definēšanai izmanto pieeju, kura tiek visbiežāk lietota, respektīvi, šķēlumu definē ar minimuma

funkciju, apvienojumu ar maksimuma funkciju un papildinājumu ar starpību. Kopu šķēlums, apvienojums un papildinājums atbilst attiecīgo izteikumu konjunkcijai, disjuncijai un negācijai.

[13]

1.5.1. Nestriktu kopu šķēlums un apvienojums

Kā divu klasisku kopu šķēlums, tā arī apvienojums var tikt vispārināts uz nestriktām kopām. Attiecīgie operatori attēlo divas nestriktas kopas uz vienu nestriktu kopu:

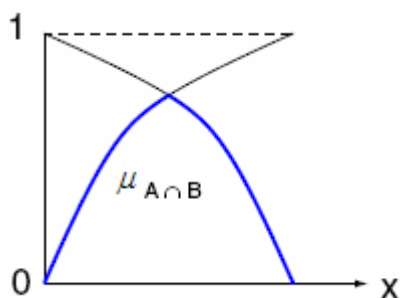
$$\cup, \cap : [0,1]^2 \rightarrow [0,1].$$

1.5.1.1. Nestriktu kopu šķēlums

Katrai funkcijai, kuru lieto kā šķēluma operatoru, ir jāpiemīt vairākām īpašībām: tai ir jābūt monotoni, komutatīvai un asociatīvai. Nestriktas kopas šķēlumam ar universālo kopu ir jābūt izejas kopai.

[12]

11. definīcija Dota universālā kopa X un nestriktas kopas A un B ar piederības funkcijām $m_A : X \rightarrow [0,1]$ un $m_B : X \rightarrow [0,1]$. Funkcija $m_{A \cap B} : X \rightarrow [0,1]$ tiek definēta ar $m_{A \cap B} = \min(m_A, m_B)$, tad $A \cap B = \{(x, m_{A \cap B}(x)) \mid x \in X\}$ sauc par nestriktu kopu A un B šķēlumu. Alternatīvais pieraksts $m_{A \cap B}(x) : m_A(x) \wedge m_B(x)$.



1.4. att. Nestriktu kopu A un B šķēlums

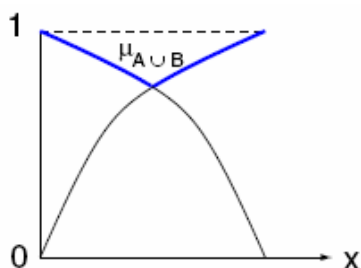
[15]

1.5.1.2. Nestriktu kopu apvienojums

Divu kopu apvienojums tiek ieviests analogi nestriktu kopu šķēlumam, bet minimuma funkcijas vietā parasti tiek lietota maksimuma funkcija. Vispārīgi katra funkcija, kurai piemīt monotonitāte, komutativitāte un asociativitāte var tikt lietota kā nestrikta apvienojuma operators, taču nestrikta kopas apvienojumam ar tukšu kopu ir jābūt sākotnējai kopai.

[12]

12. definīcija Dota universālā kopa X un divas nestrikta kopas A un B ar piederības funkcijām $m_A : X \rightarrow [0,1]$ un $m_B : X \rightarrow [0,1]$. Funkcija $m_{A \cup B} : X \rightarrow [0,1]$ ir definēta ar $m_{A \cup B} = \max(m_A, m_B)$, tad $A \cup B = \{(x, m_{A \cup B}(x)) \mid x \in X\}$ sauc par nestriktu kopu A un B apvienojumu. Alternatīvais pieraksts $m_{A \cup B}(x) : m_A(x) \vee m_B(x)$.



1.5. att. Nestriktu kopu A un B apvienojums

[15]

1.5.2. Nestriktu kopu papildinājums

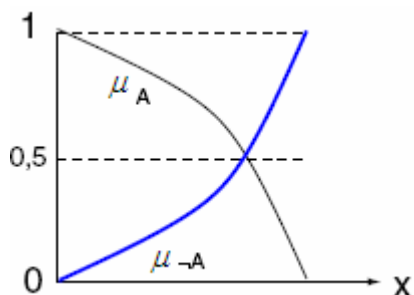
Nestrikta kopas A papildinājums vispārīgi ir definēts ar funkciju

$$c: [0,1] \rightarrow [0,1],$$

kura katrai piederības vērtībai $m_A(x)$ piešķir vērtību $c(m_A(x))$. Šī vērtība tiek interpretēta kā piederības vērtība elementam x nestriktā kopā, kura reprezentē nestrikta kopas A negāciju. Acīmredzot ir daudzi elementi, kuriem ir pozitīva piederības vērtība savā nestriktā kopā, kā arī papildinājumā.

[12]

13. definīcija Dota universālā kopa X un nestrikta kopa A ar piederības funkciju $m_A : X \rightarrow [0,1]$. Funkcija $m_{A^c} : X \rightarrow [0,1]$ ir definēta ar sakarību $m_{A^c} = 1 - m_A$, tad $A^c = \{(x, m_{A^c}(x)) | x \in X\}$ sauc par nestrikta kopas A papildinājumu. Alternatīvais pieraksts: $A^c = \neg A$, $m_{\neg A}(x) = 1 - m_A(x)$.



1.6. att. Kopas A papildinājums

Tā kā klasiskas kopas ir speciālas nestrikta kopas, tad arī nestrikta papildinājuma attēlojumam visiem $a, b \in [0, 1]$ jāizpildās sekojošiem nosacījumiem:

- $c(0) = 1$ un $c(1) = 0$ (robežnosacījumi)
- $a \leq b \Rightarrow c(a) \geq c(b)$ (monotonitāte)
- $c(c(a)) = a$ (involūcija)

[13]

Funkcijai, kuru lieto kā nestrikta papildinājumu, ir jāpiemīt sekojošai īpašībai:

pie vērtībām 0 un 1 tai jāuzvedas kā klasiskam papildinājumam, tātad $c(0)=1$ un $c(1)=0$. Funkcijai jābūt monotoni dilstošai, jo citādi lielāka piederības vērtība dotu lielāku papildinājumu, kas ir pretrunā priekšstatam par papildinājumu. Nepieciešams, lai papildinājuma papildinājums ir sākotnējā vērtība.

[12]

1.6. Nestriktu kopu operāciju īpašības

Tiek apskatītas īpašības, kuras ir spēkā klasiskajām kopām un norādīts, kuri no klasisko kopu teorijas likumiem darbojas arī nestriktām kopām un kuri ne.

Involūcija

$$(A^c)^c = A$$

Idempotence

$$A \cup A = A$$

$$A \cap A = A$$

Papildinājuma absorbcija neizpildās

$$A \cup (A^c \cap B) \neq A \cup B$$

$$A \cap (A^c \cup B) \neq A \cap B$$

De Morgāna likumi

$$(A \cap B)^c = A^c \cup B^c$$

$$(A \cup B)^c = A^c \cap B^c$$

Asociatīvā īpašība

$$(A \cup B) \cup C = A \cup (B \cup C)$$

$$(A \cap B) \cap C = A \cap (B \cap C)$$

Komutatīvā īpašība

$$A \cup B = B \cup A$$

$$A \cap B = B \cap A$$

Distributīvā īpašība

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

Teorēma par papildinājumu neizpildās

$$A \cap A^c \neq \emptyset$$

Teorēma par izslēgto trešo neizpildās

$$A \cup A^c \neq X$$

Identitāte

$$A \cup X = X$$

$$A \cap \emptyset = \emptyset$$

[10]

2. NESTRIKTAS METRIKAS UN TO PIELIETOJUMS ATTĒLU FILTRĒŠANĀ

Šī nodaļa uzrakstīta, izmantojot literatūras sarakstā minētus darbus [5], [8], [9]. Paragrāfs par nestriktām metrikām balstās uz [5] un papildus izveidotiem pierādījumiem.

Tiek sniegti jauni nestriktu metriku piemēri un pēc teorētiskā apraksta dots iepriekš apskatīto nestrikto metriku pielietojums krāsainu attēlu apstrādē. Ir aplūkota nestrikta metrika, kas reizē ņem vērā divus dažādus attāluma kritērijus starp attēla krāsainiem pikseļiem. Šī nestriktā metrika tiek lietota, lai filtrētu trokšņainus attēlus.

Pirms tiek pētīti konkrēti nestriktu metriku piemēri, tiek definēta t-norma, kas ir nepieciešams, lai raksturotu nestriktu metriku, kā arī dota nestrikta metrikas definīcija.

2.1. t-normas jēdziens

14. definīcija Par t-normu sauc attēlojumu $T : [0,1] \times [0,1] \rightarrow [0,1]$, kas apmierina sekojošus nosacījumus ($x, y, z \in [0,1]$):

1. $T(x, y) = T(y, x)$ (simetrijas nosacījums);
2. $T(T(x, y), z) = T(x, T(y, z))$ (asociativitāte);
3. $x_1 \leq x_2 \Rightarrow T(x_1, y) \leq T(x_2, y)$ (monotonitāte);
4. $T(x, 1) = x$.

Šeit definē t-normu kā funkciju, kura darbojas kopā $[0,1] \times [0,1]$ un pieņem savas vērtības kopā $[0,1]$. Tomēr apskatot konkrētas t-normas, pētot to īpašības vispār, strādājot ar t-normām, bieži ir izdevīgi līdzās „funkcionālai” t-normas definīcijai izmantot arī ekvivalentu „algebrisku” tās raksturojumu. Citiem vārdiem pieraksta $T(x, y)$ vietā bieži lieto pierakstu $x * y$, kur $*$ ir bināra algebriskā operācija kopā $[0,1]$. Tātad $T(x, y) =: x * y$.

Aksiomas 1. - 4. no t-normas definīcijas „algebriskā formā” izskatīsies šādi:

1. $x * y = y * x$ - tātad operācija $*$ ir komutatīva;
2. $(x * y) * z = x * (y * z)$ - tātad operācija $*$ ir asociatīva;
3. $x_1 \leq x_2 \Rightarrow x_1 * y \leq x_2 * y$ - tātad operācija $*$ ir monotona;
4. $x * 1 = x$ - tātad, interpretējot $*$ kā vispārinātu reizinājuma operāciju, elementam 1 ir „vieninieka loma”.

t-normu pamatpiemēri

1. min-t-norma T_M .

Šo t-normu definē ar formulu:

$$T_M(x, y) = x \wedge y$$

Citiem vārdiem t-norma T_M kopas $[0,1]$ elementiem x un y piekārto šo elementu minimumu. Tāpēc to sauc par min-t-normu.

2. Reizinājuma t-norma T_P .

Šo t-normu definē ar formulu:

$$T_P(x, y) = x \cdot y$$

Tātad šī t-norma piekārto skaitļiem x un y šo skaitļu reizinājumu.

15. definīcija t-normu $T : [0,1] \times [0,1] \rightarrow [0,1]$ sauc par nepārtrauktu, ja tā ir nepārtraukta, kā pirmā argumenta funkcija.

[9]

2.2. Nestrikta metrikas jēdziens

16. definīcija Nestrikta metriska telpa ir sakārtots trijnieks $(X, M, *)$, kurā X ir netukša kopa, $*$ ir nepārtraukta t-norma un M ir nestrikta kopa telpā $X \times X \times (0, +\infty)$. Visiem $x, y, z \in X$ un $s, t > 0$ izpildās sekojoši nosacījumi:

1. $M(x, y, t) > 0$;
2. $M(x, y, t) = 1$ tad un tikai tad, ja $x = y$;
3. $M(x, y, t) = M(y, x, t)$;
4. $M(x, y, t) * M(y, z, s) \leq M(x, z, t + s)$;
5. $M(x, y, t) : (0, +\infty) \rightarrow (0, 1]$ ir nepārtraukts pie fiksēta $x, y \in X$ kā funkcija no t .

Ja $(X, M, *)$ ir nestrikta metriska telpa, $(M, *)$ vai vienkārši M sauc par nestriktu metriku kopā X . Nestrikta versija trīsstūra nevienādībai ir ietverta 4. īpašībā. Vērtība $M(x, y, t)$ tiek uzskatīta par tuvuma pakāpi no x līdz y , ņemot vērā parametru t .

17. definīcija Nestrikta metrika M tiek saukta par stacionāru, ja tā nav atkarīga no t , t.i., katram $x, y \in X$, funkcija $M_{x,y}(t) = M(x, y, t)$ ir konstanta.

Ja M ir stacionāra nestrikta metrika, $M(x, y, t)$ vietā raksta vienkārši $M(x, y)$.

2.3. Nestrikta metriku piemēri

Visā šajā paragrāfā X ir netukša kopa, N - naturālu skaitļu kopa, R^+ - kopa no pozitīviem, reāliem skaitļiem un M ir funkcija, kas ir definēta telpā $X \times X \times R^+$ un pieņem vērtības $(0, 1]$.

1. piemērs

$f : X \rightarrow R^+$ ir injekcija un $g : R^+ \rightarrow [0, +\infty)$ ir augoša, nepārtraukta funkcija. Fiksē $a, b > 0$, definē M :

$$M(x, y, t) = \left(\frac{(\min\{f(x), f(y)\})^a + g(t)}{(\max\{f(x), f(y)\})^a + g(t)} \right)^b. \quad (2.1)$$

(M, \cdot) ir nestrikta metrika kopā X .

[5]

Pierādījums

1. $M(x, y, t) > 0$;

Tā kā funkcija f pieņem tikai pozitīvas vērtības un g nenegatīvas, ir acīmredzami, ka 1. īpašība izpildās.

2. $M(x, y, t) = 1$ tad un tikai tad, ja $x = y$;

$$M(x, y, t) = 1$$

$$\left(\frac{(\min\{f(x), f(y)\})^a + g(t)}{(\max\{f(x), f(y)\})^a + g(t)} \right)^b = 1$$

Tā kā $a, b > 0$, var veikt sekojošus pārveidojumus:

$$(\min\{f(x), f(y)\})^a + g(t) = (\max\{f(x), f(y)\})^a + g(t) \Leftrightarrow$$

$$(\min\{f(x), f(y)\})^a = (\max\{f(x), f(y)\})^a \Leftrightarrow$$

$$\min\{f(x), f(y)\} = \max\{f(x), f(y)\}$$

$$\min\{f(x), f(y)\} = \max\{f(x), f(y)\} \Leftrightarrow f(x) = f(y)$$

$$3. M(x, y, t) = M(y, x, t);$$

$$M(x, y, t) = \left(\frac{(\min\{f(x), f(y)\})^a + g(t)}{(\max\{f(x), f(y)\})^a + g(t)} \right)^b$$

$$M(y, x, t) = \left(\frac{(\min\{f(y), f(x)\})^a + g(t)}{(\max\{f(y), f(x)\})^a + g(t)} \right)^b$$

Īpašība acīmredzami izpildās.

$$4. M(x, y, t) * M(y, z, s) \leq M(x, z, t + s);$$

Pieņem, ka $f(x) \leq f(z)$, var izšķirt trīs gadījumus:

$$f(x) \leq f(y) \leq f(z) \quad f(y) \leq f(x) \leq f(z) \quad f(x) \leq f(z) \leq f(y)$$

Apskata atsevišķi katru no iespējamajiem variantiem. To pierādīšanai izmanto divus papildus rezultātus.

1. Pārbauda, vai izpildās nevienādība:

$$\frac{a+a}{b+a} \stackrel{?}{\leq} \frac{a+b}{b+b}$$

Ir zināms, ka $a, b > 0$, $a, b \geq 0$ un ir spēkā nevienādības $a \leq b$, $a \leq b$.

$$ab + ab + ab + ba \stackrel{?}{\leq} ab + aa + bb + ab$$

$$ab + ba \stackrel{?}{\leq} aa + bb$$

$$(b-a)a \stackrel{?}{\leq} (b-a)b$$

Nevienādība izpildās, jo $(b-a) \geq 0$ un $a \leq b$.

2. Pārbauda, vai izpildās nevienādība:

$$\frac{a+a}{b+a} \stackrel{?}{\leq} \frac{b+b}{a+b}$$

Ir zināms, ka $a, b > 0$, $a, b \geq 0$ un ir spēkā nevienādības $a \leq b$, $a \leq b$.

$$aa + aa + ba + ab \stackrel{?}{\leq} bb + ab + bb + ab$$

$$a(a-b) + aa \stackrel{?}{\leq} b(b-a) + bb$$

Ne vienādība izpildās, jo $(b-a) \geq 0$, $a \leq b$ un $a \leq b$.

$$4.1. \quad f(x) \leq f(y) \leq f(z);$$

$$M(x, y, t) = \left(\frac{(\min\{f(x), f(y)\})^a + g(t)}{(\max\{f(x), f(y)\})^a + g(t)} \right)^b$$

$$M(x, y, t) = \left(\frac{f(x)^a + g(t)}{f(y)^a + g(t)} \right)^b$$

$$M(y, z, s) = \left(\frac{(\min\{f(y), f(z)\})^a + g(s)}{(\max\{f(y), f(z)\})^a + g(s)} \right)^b$$

$$M(y, z, s) = \left(\frac{f(y)^a + g(s)}{f(z)^a + g(s)} \right)^b$$

$$M(x, z, t+s) = \left(\frac{(\min\{f(x), f(z)\})^a + g(t+s)}{(\max\{f(x), f(z)\})^a + g(t+s)} \right)^b$$

$$M(x, z, t+s) = \left(\frac{f(x)^a + g(t+s)}{f(y)^a + g(t+s)} \right)^b \cdot \left(\frac{f(y)^a + g(t+s)}{f(z)^a + g(t+s)} \right)^b$$

Jāpārbauda, vai izpildās nevienādība

$$M(x, y, t) \cdot M(y, z, t) \leq M(x, z, t+s)$$

$$\left(\frac{f(x)^a + g(t)}{f(y)^a + g(t)} \right)^b \cdot \left(\frac{f(y)^a + g(s)}{f(z)^a + g(s)} \right)^b \stackrel{?}{\leq} \left(\frac{f(x)^a + g(t+s)}{f(y)^a + g(t+s)} \right)^b \cdot \left(\frac{f(y)^a + g(t+s)}{f(z)^a + g(t+s)} \right)^b$$

Izmantojot iepriekš iegūto 1. rezultātu, var secināt, ka

$$\left(\frac{f(x)^a + g(t)}{f(y)^a + g(t)} \right)^b \leq \left(\frac{f(x)^a + g(t+s)}{f(y)^a + g(t+s)} \right)^b$$

$$\left(\frac{f(y)^a + g(s)}{f(z)^a + g(s)}\right)^b \leq \left(\frac{f(y)^a + g(t+s)}{f(z)^a + g(t+s)}\right)^b$$

Tas nozīmē, ka dotā īpašība izpildās.

$$4.2. \quad f(y) \leq f(x) \leq f(z);$$

$$M(x, y, t) = \left(\frac{f(y)^a + g(t)}{f(x)^a + g(t)}\right)^b$$

$$M(y, z, s) = \left(\frac{f(y)^a + g(s)}{f(z)^a + g(s)}\right)^b$$

$$M(x, z, t+s) = \left(\frac{f(x)^a + g(t+s)}{f(y)^a + g(t+s)}\right)^b \cdot \left(\frac{f(y)^a + g(t+s)}{f(z)^a + g(t+s)}\right)^b$$

$$\left(\frac{f(y)^a + g(t)}{f(x)^a + g(t)}\right)^b \cdot \left(\frac{f(y)^a + g(s)}{f(z)^a + g(s)}\right)^b \leq \left(\frac{f(x)^a + g(t+s)}{f(y)^a + g(t+s)}\right)^b \cdot \left(\frac{f(y)^a + g(t+s)}{f(z)^a + g(t+s)}\right)^b$$

Izmantojot 1. un 2. rezultātus, var secināt, ka

$$\left(\frac{f(y)^a + g(t)}{f(x)^a + g(t)}\right)^b \leq \left(\frac{f(x)^a + g(t+s)}{f(y)^a + g(t+s)}\right)^b$$

$$\left(\frac{f(y)^a + g(s)}{f(z)^a + g(s)}\right)^b \leq \left(\frac{f(y)^a + g(t+s)}{f(z)^a + g(t+s)}\right)^b$$

Tas nozīmē, ka dotā īpašība izpildās.

$$4.3. \quad f(x) \leq f(z) \leq f(y);$$

$$M(x, y, t) = \left(\frac{f(x)^a + g(t)}{f(y)^a + g(t)}\right)^b$$

$$M(y, z, s) = \left(\frac{f(z)^a + g(s)}{f(y)^a + g(s)}\right)^b$$

$$M(x, z, t+s) = \left(\frac{f(x)^a + g(t+s)}{f(y)^a + g(t+s)} \right)^b \cdot \left(\frac{f(y)^a + g(t+s)}{f(z)^a + g(t+s)} \right)^b$$

$$\left(\frac{f(x)^a + g(t)}{f(y)^a + g(t)} \right)^b \cdot \left(\frac{f(z)^a + g(s)}{f(y)^a + g(s)} \right)^b \stackrel{?}{\leq} \left(\frac{f(x)^a + g(t+s)}{f(y)^a + g(t+s)} \right)^b \cdot \left(\frac{f(y)^a + g(t+s)}{f(z)^a + g(t+s)} \right)^b$$

Izmantojot 1. un 2. rezultātus, var secināt, ka

$$\left(\frac{f(x)^a + g(t)}{f(y)^a + g(t)} \right)^b \leq \left(\frac{f(x)^a + g(t+s)}{f(y)^a + g(t+s)} \right)^b$$

$$\left(\frac{f(z)^a + g(s)}{f(y)^a + g(s)} \right)^b \leq \left(\frac{f(y)^a + g(t+s)}{f(z)^a + g(t+s)} \right)^b$$

Īpašība ir pierādīta gadījumā, ja $f(x) \leq f(z)$. Analogiski var pārliecināties, ka īpašība izpildās arī visos pārējos gadījumos.

5. $M(x, y, t): (0, +\infty) \rightarrow (0, 1]$ ir nepārtraukts;

$$M(x, y, t) = \left(\frac{(\min\{f(x), f(y)\})^a + g(t)}{(\max\{f(x), f(y)\})^a + g(t)} \right)^b$$

Īpašība ir spēkā, jo g ir nepārtraukta funkcija.

Ja f definē kā identisko funkciju, g kā konstantu funkciju $g(t) = k$, parametri $a = b = 1$ un kopa $X = (-k, +\infty)$, $k > 0$, tad no (2.1) kā speciālu gadījumu iegūst:

$$M(x, y) = \frac{\min\{x, y\} + k}{\max\{x, y\} + k}. \quad (2.2)$$

2. piemērs

18. definīcija Metrika kopā X ir funkcija $d: X \times X \rightarrow R$. Visiem $x, y, z \in X$ šī funkcija apmierina sekojošās īpašības:

1. $d(x, y) \geq 0$;
2. $d(x, y) = 0 \Leftrightarrow x = y$;
3. $d(x, y) = d(y, x)$;
4. $d(x, z) \leq d(x, y) + d(y, z)$.

$g : R^+ \rightarrow R^+$ ir augoša un nepārtraukta funkcija un d ir metrika kopā X , parametrs $m > 0$.

Definē funkciju M :

$$M(x, y, t) = \frac{g(t)}{g(t) + m \cdot d(x, y)}. \quad (2.3)$$

(M, \cdot) ir nestrikta metrika kopā X .

[5]

Pierādījums

1. $M(x, y, t) > 0$;

Īpašība izpildās, jo funkcija g pieņem tikai pozitīvas vērtības, parametrs $m > 0$ un $d(x, y) \geq 0$ $\forall x, y \in X$.

2. $M(x, y, t) = 1$ tad un tikai tad, ja $x = y$;

$$M(x, y, t) = 1$$

$$\frac{g(t)}{g(t) + m \cdot d(x, y)} = 1$$

$$g(t) = g(t) + m \cdot d(x, y)$$

$$m \cdot d(x, y) = 0$$

Tā kā $m > 0$, tad $d(x, y) = 0$. No metrikas īpašībām zināms, ka $d(x, y) = 0 \Leftrightarrow x = y$.

3. $M(x, y, t) = M(y, x, t)$;

$$M(x, y, t) = \frac{g(t)}{g(t) + m \cdot d(x, y)}$$

$$M(y, x, t) = \frac{g(t)}{g(t) + m \cdot d(y, x)}$$

Īpašība ir spēkā, jo $d(x, y) = d(y, x)$.

4. $M(x, y, t) * M(y, z, s) \leq M(x, z, t + s)$;

$$\frac{g(t)}{g(t) + m \cdot d(x, y)} \cdot \frac{g(s)}{g(s) + m \cdot d(y, z)} \leq \frac{g(t+s)}{g(t+s) + m \cdot d(x, z)}$$

Vienkāršības pēc apzīmē:

$$g(t) = k$$

$$g(s) = l$$

$$g(t+s) = d$$

$$m \cdot d(x, y) = a$$

$$m \cdot d(y, z) = b$$

$$m \cdot d(x, z) = c$$

No metrikas īpašībām ir zināms, ka $c \leq a + b$.

$$\frac{k \cdot l}{(k+a)(l+b)} \stackrel{?}{\leq} \frac{d}{d+c}$$

$$kld + klc \stackrel{?}{\leq} kld + ald + bkd + abd$$

$$klc - (ald + bkd) \stackrel{?}{\leq} abd$$

Apskata lielumu nevienādības kreisajā pusē un samazina reizinātājus ld un kd , aizstājot tos ar kl .

$$kl(c - (a + b)) \leq 0 \Rightarrow kl(c - (a + b)) \leq abd \Rightarrow klc - (ald + bkd) \leq abd$$

5. $M(x, y, t): (0, +\infty) \rightarrow (0, 1]$ ir nepārtraukts;

$$M(x, y, t) = \frac{g(t)}{g(t) + m \cdot d(x, y)}$$

Funkcija g ir nepārtraukta un pieņem pozitīvas vērtības, parametrs $m > 0$ un $d(x, y) \geq 0$ $\forall x, y \in X$.

Definē $g(t) = t^n$, kur $n \in \mathbb{N}$ un $m = 1$, tad iegūst īpašu gadījumu un (2.3) pārņem formā:

$$M(x, y, t) = \frac{t^n}{t^n + d(x, y)}. \quad (2.4)$$

2.4. Attēlu filtrēšana, izmantojot nestriktās metrikas

Nestriktās metrikas sniedz virkni priekšrocību salīdzinājumā ar klasiskajām metrikām, kas motivē to izpēti praktiskām problēmām. Divas svarīgākās priekšrocības salīdzinājumā ar klasiskajām metrikām: pirmkārt, vērtības, kuras pieņem nestriktās metrikas, atrodas intervālā $(0,1]$, neskatoties uz datiem un kā tiek definēts attālums starp jēdzieniem, kurus mēra. Tas norāda, ka ir vienkārši kombinēt dažādus distances kritērijus, kas var oriģināli atrasties diezgan dažādos apgabalos, bet nestriktā metrika noved pie kopēja intervāla. Šādā veidā dažādu distances kritēriju kombinācija var tikt veikta tiešā veidā. Otrkārt, nestrikta metrikas perfekti saskan ar citu nestriktu tehniku pielietojumu, tā kā vērtība, kuru sniedz nestrikta metrika var tikt tieši izmantota vai interpretēta kā nestrikta pārliecības pakāpe. Tas sniedz iespēju tieši ietvert nestriktu metriku kā daļu no citām kompleksām nestriktām sistēmām.

Ilustrējot nestrikto metriku priekšrocības, tiek apskatīts nestrikto metriku pielietojums krāsainu attēlu filtrēšanai, kur tiek kombinētas divas dažādas nestrikta metrikas vienā unikālā metrikā.

Apzīmē ar F digitālu attēlu, kuru vēlas apstrādāt. Attēlu apstrādā, pielietojot *slīdošā loga* pieeju. Lai apskatītu katru pikseli, ņem pikselus kvadrātiskā blokā W ar izmēru $n \times n$, kur $n > 1$ un nepāra skaitlis. Aplūkotais pikselis atrodas bloka centrā, attiecīgi blakus esošie pikseļi blokā ap to. Ar F_q apzīmē n^2 krāsu vektorus, kas ir asociēti ar pikseļiem logā W , kur F_c norāda krāsu vektoru, kas ir asociēts ar konkrēto apskatīto pikseli. Krāsains pikselis ir raksturots ar tā atrašanās vietu attēlā $q = \{q_1, q_2\}$, kas ir dota ar koordinātēm, un ar trim RGB krāsu komponentēm, t.i., $F_q = \{F_q^R, F_q^G, F_q^B\}$. RGB attēlos attālumu starp diviem krāsas vektoriem var mērīt ar līdzību starp to krāsu komponentēm un arī telpisko tuvumu starp pikseļiem. Labu rezultātu iegūšanai lieto nestriktu metriku, kas vienlaicīgi reprezentē abus distances kritērijus. Izmanto nestrikto metriku R , lai mērītu līdzību starp krāsu vektoriem F_i un F_j .

$$R(F_i, F_j) = \prod_{l=1}^3 \frac{\min\{F_i^l, F_j^l\} + K}{\max\{F_i^l, F_j^l\} + K}. \quad (2.5)$$

Šī nestriktā metrika ir ieviesta (2.2) vienādībā un var pierādīt, ka tā tiešām ir nestrikta metrika, tā kā nestriktu metriku reizinājums arī ir metrika, skat. [8]. Šī nestriktā metrika uzvedas tā, ka

diviem dažādiem pāriem no viena otrai sekojošām vērtībām (vai vektoriem) nestriktā distance, kas ir dota ar nestrikto metriku, ir dažāda. Šo efektu var mazināt, paaugstinot parametra K vērtību vienādībā (2.5). No iepriekšējiem eksperimentāliem rezultātiem var secināt, ka ļoti augstas K vērtības nevajadzētu lietot, lai izvairītos no nestriktās metrikas jūtīguma paaugstināšanās. Optimālā K vērtība ir atkarīga gan no attēla, gan trokšņa blīvuma. RGB krāsainajiem attēliem ir atrasts, ka vērtības intervālā [512,2048] ir piemērotas. Eksperimentālā ceļā ir izvēlēta vērtība $K = 1024$, jo šī heksadecimālā operācija ir aprēķiniem vienkārša.

Telpiskais tuvums starp diviem attēla pikseļiem F_i un F_j ir dots ar nestrikto metriku S :

$$S(F_i, F_j, t) = \frac{t}{t + \|i - j\|}. \quad (2.6)$$

Attiecīgi $\|\cdot\|$ norāda Eiklīda normu. Šī metrika ir speciāls gadījums no (2.4) vienādojuma, kad $n = 1$. t ir parametrs, kas ir spējīgs izlīdzināt nestriktās metrikas jūtīgumu. Konstruē reizinājumu no iepriekš apskatītajām nestriktajām metriķām R un S vienā nestriktā metriķā C :

$$C(F_i, F_j, t) = R(F_i, F_j)S(F_i, F_j, t). \quad (2.7)$$

Tādā veidā nestriktā metrika C mēra tuvumu starp diviem attēla pikseļiem, ņemot vienlaicīgi vērā līdzību starp krāsu komponentēm un telpisko tuvumu starp pikseļiem.

Digitāla attēla filtrēšanas problēma sevī ietver noņemt nekorektos trokšņainos pikseļus un aizstāt tos ar beztrokšņu pikseļiem. Ar slīdošā loga pieeju to var veikt, identificējot blokā W oriģinālo F_c , lai aizstātu ar beztrokšņu pikseli \hat{F}_c .

Vektoru kontekstā šī identifikācija var tikt izveidota sekojoši - beztrokšņa vektors \hat{F}_c ir identificēts kā vektors blokā W , kas maksimizē uzkrājumu no piemērota tuvuma mēra, ņemot vērā visus pārējos pikseļus W . Šajā pielietojumā izmanto nestrikto metriķu C kā mēru, kuru maksimizē. Tā kā šī nestriktā metrika ņem vērā abas - līdzību starp krāsu komponentēm, kā arī telpisko tuvumu starp pikseļiem attēlā - \hat{F}_c būs identificēts kā vektors blokā W , kas vienlaicīgi ir vislīdzīgākais un telpiski tuvākais visiem pārējiem vektoriem W . Detalizēti apraksta \hat{F}_c noteikšanas procedūru.

Katram vektoram F_k filtra logā ir jāaprēķina uzkrātu mēru attiecībā pret visiem pārējiem vektoriem logā, lai izveidotu dilstošu sakārtojumu:

$$A_k = \sum_{j \in W, j \neq k} C(F_k, F_j, t).$$

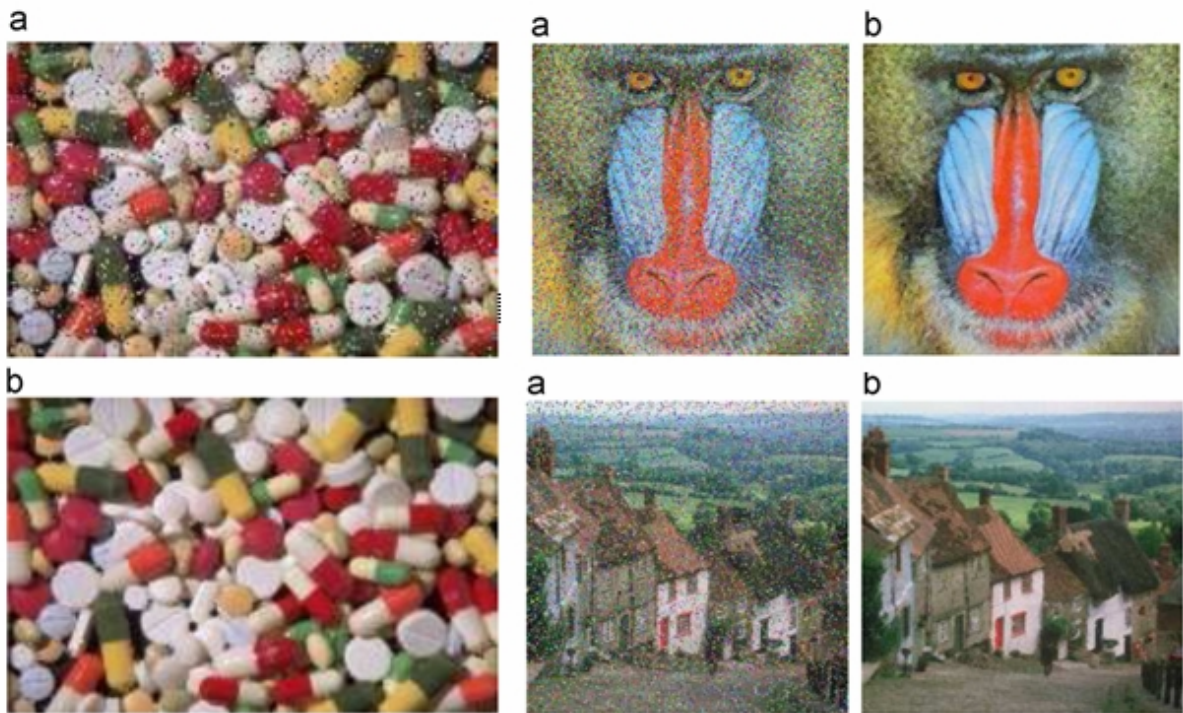
A_k vērtības sakārto dilstošā virknē (vērtība, kura atrodas r -tajā pozīcijā tiek apzīmēta ar $A_{(r)}$):

$$A_{(0)} \geq A_{(1)} \geq \dots \geq A_{(n^2-1)}$$

Šis sakārtojums norāda tādu pašu vektoru F_k sakārtojumu:

$$F_{(0)} \geq F_{(1)} \geq \dots \geq F_{(n^2-1)}$$

Tad filtra izvade \hat{F}_c būs vektors $F_{(0)}$, kas ieņem zemāko pakāpi sakārtotā virknē. Filtra izvade vienmēr ir viens no vektoriem, kas atrodas filtra logā. Tādā veidā no filtra nevar tikt ieviestas nepareizas krāsas.



2.1. att. Ar nestrikto metriku filtru iegūtie rezultāti

No attēliem var redzēt, ka troksnis tiek reducēts. Šis rezultāts tiek sasniegts, jo izvades vektors katrā filtrēšanas logā ir noteikts kā vektors, kas ir telpiski tuvs visiem pārējiem vektoriem logā. Tādā veidā izvairās no iespējas aizstāt pikseli ar citu, kas atrodas tālu no tā, kas nav piemēroti, lai saglabātu detaļas un šķautnes.

Telpiskā kritērija svarīgums var tikt izlīdzināts, modificējot t parametra vērtību. Augstākas t vērtības samazina kombinētās nestriktās metrikas jūtīgumu pret telpisko kritēriju un paaugstina filtra trokšņa redukcijas spēju, bet zemākas t vērtības piešķir vairāk svarīgumu telpiskajam kritērijam un tā labāku oriģinālā attēla saglabāšanu. Tas nozīmē, ka augstākas t vērtības ir piemērotas augstākam trokšņa blīvumam, tā kā šajos gadījumos ir svarīgāk filtra robustums nekā oriģinālu detaļu saglabāšana. Šo var aplūkot 1. tabulā, kur parādīts dažādiem trokšņu procentuāliem intervāls no iespējamām vērtībām t parametram, kas var sniegt labu rezultātu. Šīs optimālās vērtības ir sameklētas eksperimentālā ceļā, lietojot testa attēlus, aptraipītus ar dažādiem trokšņa procentiem. Attiecīgi meklē vērtības, kas maksimizē peak signal noise ratio (PSNR) kvalitātes mēru.

19. definīcija PSNR mērs tiek definēts ar vidējo kvadrātisko kļūdu (*angļu val.* mean squared error, *saīs.* MSE), kas diviem $m \times n$ izmēra attēliem I un K , kur viens no attēliem tiek uzskatīts kā oriģinālais un otrs tā trokšņainā aproksimācija, ir definēts kā:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2.$$

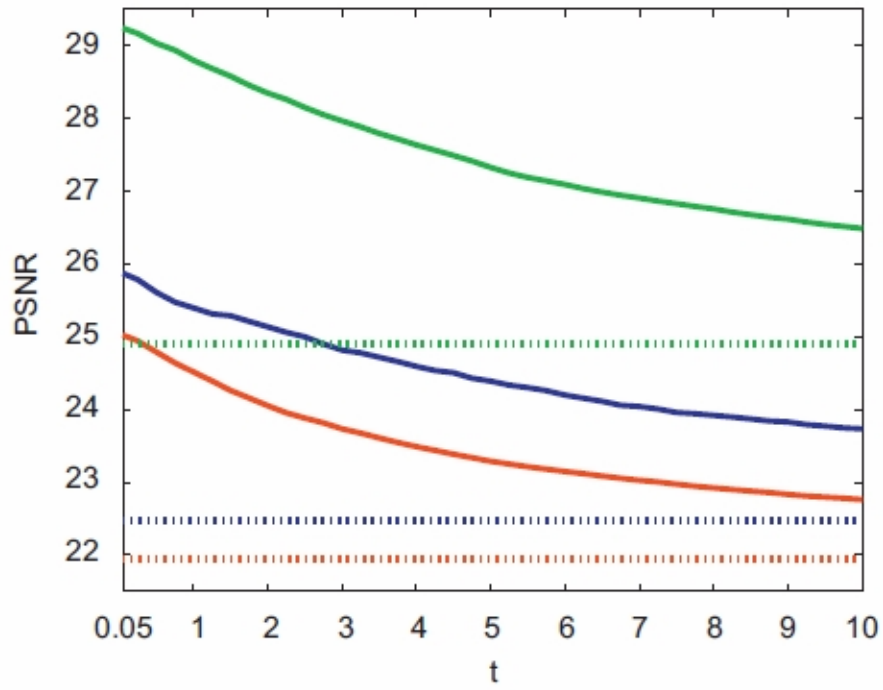
PSNR ir definēts sekojoši:

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) = \\ &= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right). \end{aligned}$$

MAX_I ir maksimālā iespējamā pikseļa vērtība attēlā.

2.1. tabulā var novērot, ka optimālā t vērtība palielinās, kad trokšņa daudzums kļūst lielāks. Tajā redzama PSNR vērtība dažādiem izvades attēliem kā funkcija no t . Augstas PSNR vērtības ir arī sasniegtas ar ļoti zemām t vērtībām ($t < 0.1$). Lietojot šādu zemu t vērtību, tiek paaugstināts telpiskā kritērija svarīgums un tā filtra spēja attēla saglabāšanai, kas noved pie augstas PSNR vērtības. Kāds troksnis tomēr eksistē, kas nozīmē, ka šie rezultāti nav piemēroti no vizuālā kvalitātes skata punkta. Tas notiek ne tikai ar PSNR, bet arī ar citiem attēla kvalitātes rādītājiem. Šī iemesla dēļ iesaka lietot nedaudz augstākas t vērtības, kas sniedz labāku trokšņa redukciju un tādā veidā vizuāli patīkamus izvades attēlus.

2.1. tabula PSNR vērtība atkarībā no t



[5]

3. MARKOVA GADĪJUMA LAUKI UN NOSACĪTIE GADĪJUMA LAUKI

Izmantotie avoti [1], [2], [3], [4], [6], [7], [14], pamatā informācija ņemta no [7].

Nodaļā vispirms tiek aplūkotas fizikālās sistēmās, uz kuru bāzes vēlāk tiek izveidoti attēla segmentāciju algoritmi – Markova gadījuma lauks (*angļu val.* Markov random field, *saīs.* MRF) un nosacītais gadījuma lauks (*angļu val.* conditional random field, *saīs.* CRF). Tiek definēti nepieciešamie jēdzieni, lai apskatītu minētos modeļus un izprastu to darbību. Paskaidrots veids, kā meklē optimālo atrisinājumu un tam nepieciešamie palīg līdzekļi, piemēram, enerģijas funkcija. Kā piemērs attēlu apstrādei ir dots *Ising* modelis.

3.1. Statistiskā fizika no neizolētām sistēmām

Šajā paragrāfā tiks apskatītas sistēmas no daļiņām (atomiem, molekulām), kas atrodas kontaktā ar siltuma enerģiju. Turpmākajā darbā tiek izveidota analogija starp attēliem un statistiskām mehāniskām sistēmām. Pikseļa krāsu līmeņi un šķautņu orientācija tiek uzskatīti kā atomu vai molekulu stāvokļi režģī kā fizikālā sistēmā.

3.1.1. Bolemana sadalījums

20. definīcija Mikrostāvoklis ir sistēmas stāvoklis, kas ir definēts ar visu atsevišķu daļiņu pašreizējo uzvedību.

Dota sistēma ar fiksētu kopējo enerģiju E , t.i., enerģija sistēmai netiek pievienota, ne arī atņemta. Kopējā enerģija ir visu atsevišķo daļiņu enerģijas summa:

$E = \sum_{i=0}^{\infty} n_i e_i$, kur n_i ir daļiņu skaits, kurām piemīt enerģija e_i . Kopējais daļiņu skaits N arī ir

konstants, $N = \sum_{i=0}^{\infty} n_i$.

Ja sistēma ir kontaktā ar siltuma enerģiju pie temperatūras T , varbūtība, ka tā ir *itajā* mikrostāvoklī ar enerģiju E_i ir dota ar Bolcmana sadalījumu:

$$p_i = \frac{e^{-E_i/k_B T}}{Z}, \quad i = 0, \dots, \infty,$$

kur k_B ir Bolcmana konstante, $k_B = 1.38 \times 10^{-23}$ J/K, un E_i ir enerģija no *itā* enerģijas stāvokļa. Z ir normalizējošā konstante un tiek atrasta pie nosacījuma, ka varbūtība, ka sistēma atrodas kādā mikrostāvoklī ir 1: $\sum_{j=0}^{\infty} p_j = 1$, tātad $Z = \sum_{j=0}^{\infty} e^{-E_j/k_B T}$.

3.1.2. Sistēmas ar mainīgu daļiņu skaitu

Iepriekšējā nodaļā tika ņemtas vērā sistēmas kontaktā ar siltuma enerģiju pie temperatūras T . Tagad aplūko sistēmas, kas ir kontaktā ar siltuma enerģiju pie temperatūras T un atrodas arī difūzā kontaktā ar daļiņu rezervuāru, kuram piemīt ķīmiskais potenciāls m .

Ķīmiskais potenciāls m raksturo vielas spēju:

- Reaģēt ar citām vielām (ķīmiska reakcija);
- Pāriet citā stāvoklī (fāzes pāreja);
- Izplesties telpā (difūzija).

Šajā gadījumā, ja daļiņu skaits var mainīties, Bolcmana sadalījums tiek modificēts un saukts par Gibbsa sadalījumu.

3.1.2.1. Gibbsa sadalījums

Tiek apskatītas sistēmas, kas ir kontaktā ar siltuma enerģiju pie temperatūras T un arī daļiņu rezervuāru, kuram ir ķīmiskais potenciāls m . Tiek meklēta varbūtība, ka sistēma atrodas konkrētā mikrostāvoklī i ar enerģiju E_i un daļiņu skaitu N_i .

$$p_i = \frac{e^{(mN_i - E_i)/k_B T}}{Z}, \quad i = 0, \dots, \infty.$$

$$Z = \sum_{j=0}^{\infty} e^{(mN_j - E_j)/k_B T}$$

Lietojot Bolcmana sadalījumu, atbrīvojas no ierobežojuma, ka visu daļiņu kopējai enerģijai ir jāsummējas uz dotu kopēju enerģiju, un atļauj uztvert katru daļiņu neatkarīgi. Lietojot Gibbsa sadalījumu, atbrīvojas no ierobežojuma, ka kopējam daļiņu skaitam katrā enerģijas līmenī ir jāsummējas uz fiksētu kopēju skaitu, un atļauj apskatīt katru enerģijas līmeni neatkarīgi.

[14]

3.2. Apzīmējuma piešķiršana attēlu analīzē

Daudzas attēlu interpretāciju un analīzes problēmas var tikt uzdotas kā apzīmējuma piešķiršanas problēmas, kurās atrisinājums problēmai ir kopa no apzīmējumiem, kas tiek piešķirti attēla pikseliem.

3.2.1. Mezgli un apzīmējumi

Apzīmējuma piešķiršanas problēma ir definēta ar kopu, kas sastāv no mezgliem, un kopu, kas sastāv no apzīmējumiem. Mezgls bieži attēlo punktu vai reģionu Eiklīda telpā, piemēram, attēla pikseli vai attēla pazīmi. S ir diskrēta kopa no m mezgliem

$S = \{1, \dots, m\}$, kur $1, \dots, m$ ir indeksi, kas norāda konkrētu pikseli.

Kopa no mezgliem var tikt kategorizēta pēc to „regularitātes”. Mezgli režģī tiek uzskatīti par telpiski *regulāriem*. Taisnstūra režģis 2D attēliem ar izmēru $n \times n$ var tikt raksturots sekojoši:

$$S = \{(i, j) | 1 \leq i, j \leq n\}.$$

Tādā veidā elementi atbilst atrašanās vietai, kurā attēls tiek apstrādāts. Mezgli, kuri nereprezentē telpisku regularitāti, tiek uzskatīti par *neregulāriem*. Savstarpējās attiecības starp mezgliem tiek aprakstītas ar *kaimiņu sistēmu* (šis jēdziens tiks paskaidrots 3.4.1.1.).

$n \times n$ izmēra attēlam pikselis (i, j) var tikt atbilstošā veidā apzīmēts ar vienu skaitli k , kur k pieņem vērtības no intervāla $\{1, 2, \dots, m\}$, kur $m = n \times n$.

Apzīmējums ir notikums, kuru var piešķirt mezglam. L ir kopa, kas sastāv no apzīmējumiem. Apzīmējumu kopa var tikt kategorizēta kā nepārtraukta vai diskrēta. Nepārtrauktajā gadījumā, apzīmējumu kopa var atbilst reālajai taisnei vai kompakam intervālam no tās.

$$L_c = [X_l, X_h] \subset R.$$

Diskrētajā gadījumā apzīmējums pieņem diskrētu vērtību no kopas L_d , kura satur M apzīmējumus:

$$L_d = \{l_1, \dots, l_M\} \text{ vai vienkārši } L_d = \{1, \dots, M\}.$$

Papildus nepārtrauktībai vēl viena apzīmējumu kopas svarīga īpašība ir apzīmējumu sakārtojums. Piemēram, elementi nepārtrauktā apzīmējumu kopā R (reālā telpa) var tikt

sakārtoti ar relāciju „mazāks nekā”. Diskrētas kopas gadījumā, aplūko sakārtotu RGB vērtību kopu. Intervāls $\{0, \dots, 255\}$ reprezentē krāsu intensitātes vērtības, kuras ir sakārtotas $0 < 1 < 2 < \dots < 255$.

3.2.2. Apzīmējuma piešķiršanas problēma

Apzīmējuma piešķiršanas problēma ir piemeklēt apzīmējumu no kopas L katram mezglam režģī S . Attēla segmentācijas problēmā jāatrod pareizo apzīmējumu f_i no kopas L mezglam $i \in S$, kur elementi kopā S ir attēla pikseļi.

Kopa $f = \{f_1, \dots, f_m\}$ tiek saukta par izveidoto attēlu no mezgliem kopā S , ņemot vērā apzīmējumus L . Kad katram mezglam ir pievienots unikāls apzīmējums $f_i = f(i)$, funkcija f var tikt uzskatīta kā funkcija ar definīcijas apgabalu S un vērtību apgabalu L :

$$f : S \rightarrow L.$$

Apzīmējuma piešķiršana mezglam var tikt uztverta kā attēlojums no kopas, kas sastāv no mezgliem režģī S , uz kopu L , kas sastāv no apzīmējumiem. Gadījuma lauku terminoloģijā izveidotais attēls tiek saukts arī par konfigurāciju.

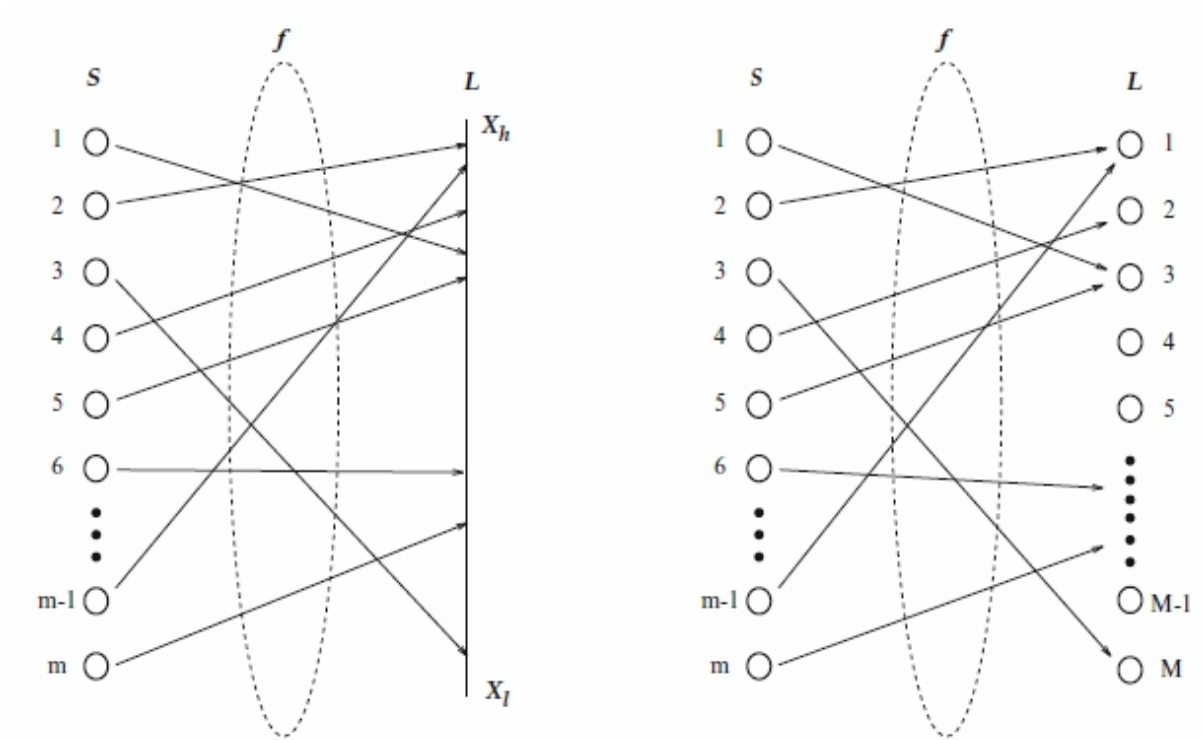
Ja katram mezglam ir tā pati apzīmējumu kopa L , tad kopa no visiem iespējamajiem attēliem (t.i., konfigurācijas telpa) ir Dekarta reizinājums

$$F = \underbrace{L \times L \times \dots \times L}_m = L^m,$$

m reizes

kur m ir attiecīgi S izmērs. Piemēram, attēla atjaunošanai kopa L satur pieļaujamās pikseļa vērtības, kuras ir kopīgas visiem pikseļa mezgliem režģī S , un F definē visus pieļaujamos attēlus.

Nepārtrauktajā gadījumā, kad $L = R$ ir reāla taisne, $F = R^m$ ir m dimensionāla reāla telpa. Kad L ir diskrēta kopa, F izmērs ir kombinatorisks. Problēmai ar m mezgliem un M apzīmējumiem, piemēram, pavisam kopā eksistē M^m iespējamo konfigurāciju skaits kopā F . Nepārtrauktai apzīmējuma piešķiršanas problēmai ir bezgalīgs konfigurāciju skaits. Starp visiem izveidotajiem attēliem toties ir tikai mazs skaits no tiem, kuri ir labi atrisinājumi, un varbūt tikai daži ir optimāli, ņemot vērā kādu papildus kritēriju.



3.1. att. Attēlojums ar nepārtrauktu apzīmējumu kopu (kreisā puse) un diskretu apzīmējumu kopu (labā puse)

3.2.3. Apzīmējuma piešķiršana pie dotiem ierobežojumiem

Informācija par tuvākajiem kaimiņiem tiek lietota, lai aprēķinātu nosacītās varbūtības. Doti ierobežojumi var tikt izteikti lokāli ar nosacītām varbūtībām $P(f_i | \{f_{i'}\})$, kur $\{f_{i'}\}$ norāda apzīmējumu kopu, kas sastāv no pārējiem mezgliem $i' \neq i$, vai globāli ar kopējo varbūtību $P(f)$.

Kopējā varbūtība parasti tiek izteikta ar lokālām varbūtībām, jo tās ir tieši novērojamas. Situācijās, kad apzīmējumi ir neatkarīgi viens no otra, kopējā varbūtība ir reizinājums no lokālām varbūtībām

$$P(f) = \prod_{i \in S} P(f_i). \quad (3.1)$$

Augšējā vienādība ietver nosacīto neatkarību

$$P(f_i | \{f_{i'}\}) = P(f_i) \quad i' \neq i. \quad (3.2)$$

Šī iemesla dēļ globāls attēls f var tikt aprēķināts, apskatot katru apzīmējumu f_i lokāli. Tam ir priekšrocība problēmu risināšanai.

Ja tiek doti ierobežojumi, apzīmējumi ir savstarpēji atkarīgi. Vienkāršās attiecības, kuras bija spēkā (3.1) un (3.2), vairs nevar izmantot. Kā aprēķināt kopējo varbūtību, lietojot nosacītās varbūtības, kļūst par netriviālu uzdevumu. MRF teorija, kura tiks apskatīta 3.4. nodaļā, sniedz matemātisku pamatojumu, lai atrisinātu šo problēmu.

3.3. Optimālā atrisinājuma meklēšana attēlu apstrādē

Optimizācijai ir svarīga loma attēlu analīzē. Problēma tiek formulēta, kā optimizēt kādu kritēriju, tieši vai netieši. Galvenais iemesls plašajam optimizācijas pielietojumam ir nenoteiktību eksistence katrā attēlu apstrādes procesā. Trokšņi un citi kvalitātes pazeminoši faktori ir šo nenoteiktību avoti.

Minēto apstākļu dēļ retos gadījumos ir iespējams iegūt precīzu vai perfektu atrisinājumu. Tā vietā vairums tiek meklēti neprecīzi, bet optimāli (kādā aspektā) atrisinājumi ar mērķa funkcijas palīdzību. Mērķa funkcija attēlo atrisinājumu kā reālu skaitli, kas mēra atrisinājuma kvalitāti kāda „labuma” vai „izmaksu” veidā. Mērķa funkcijas formulējums nosaka, cik dažādi ierobežojumi pastāv, tādi var būt pikseļa īpašības. Piemēram, krāsu intensitāte un attiecības starp pikseļiem var tikt iekodētas funkcijā. Formulējums definē optimālo atrisinājumu.

Attēlu apstrādē tiek meklēts varbūtības $P(f)$ maksimums, attiecīgi ir jāminimizē mērķa funkciju, kas ir uzdota enerģijas funkcijas formā. Optimizācijas lomas dēļ ir svarīgi pētīt attēlu analīzes problēmas no optimizācijas skata punkta un attīstīt metodoloģijas optimizācijas balstītai modelēšanai.

3.4. Matemātiski MRF modeļi

Šī nodaļa iepazīstina ar MRF teorijas pamatiem un apraksta svarīgus matemātiskus MRF modeļus, lai ilustrētu attēlu pazīmes.

3.4.1. Markova gadījuma lauks un Gibsa sadalījums

MRF ir viens no apskates objektiem varbūtību teorijā, lai analizētu fizikālu parādību telpiskas vai kontekstuālas atkarības. Tas tiek lietots attēlu apstrādē, lai pamatotu apzīmējumu savstarpējo mijiedarbību varbūtību sadalījumus. Šī apakšnodaļa iepazīstina ar apzīmējumiem un rezultātiem, kas ir attiecināmi uz MRF.

3.4.1.1. Kaimiņu sistēma un klišes

Mezgli režģī S ir saistīti viens ar otru, un šīs saistības apraksta kaimiņu sistēma. Kaimiņu sistēma režģim S ir definēta sekojoši

$$N = \{N_i | \forall i \in S\},$$

kur N_i ir kopa no mezgliem, kas ir kaimiņi mezglam i . Kaimiņu attiecībām ir sekojošās īpašības:

1. Mezgls nav kaimiņš pats sev: $i \notin N_i$.
2. Kaimiņu attiecības ir abpusējas: $i \in N_{i'} \Leftrightarrow i' \in N_i$.

Regulāram režģim S kaimiņu kopa mezglam i ir definēta kā kopa, kas sastāv no mezgliem ar rādiusu \sqrt{r}

$$N_i = \{i' \in S \mid \|dist(pixel_{i'}, pixel_i)\|^2 \leq r, i' \neq i\}.$$

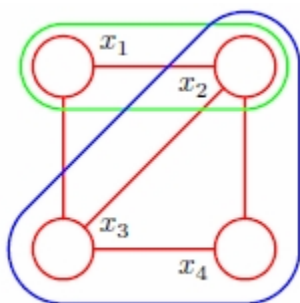
Parametrs r pieņem naturālu vērtību.

Pirmās kārtas kaimiņu sistēmā, arī saukta par 4-kaimiņu sistēmu, katram (iekšējam) mezglam ir četri kaimiņi kā parādīts 3.3. att. (a). Attiecīgi X norāda mezglu, kuru aplūko, un nulles ir tā kaimiņi. Otrās kārtas kaimiņu sistēmā, arī saukta par 8-kaimiņu sistēmu, katram (iekšējam) mezglam ir astoņi kaimiņi kā parādīts 3.3. att. (b). Skaitļi $n = 1, \dots, 5$, kas parādīti 3.3. att. (c) norāda visattālākos mezglus, kuri ir kaimiņi, n -tās kārtas kaimiņu sistēmās.

Kad elementu sakārtojums kopā S ir specificēts, kaimiņu kopa var tikt determinēta precīzāk. Piemēram, ja $S = \{1, \dots, m\}$ ir sakārtota mezglu kopa un tās elementi atbilst pikseļiem 1D attēlā, iekšējam mezglam $i \in \{2, \dots, m-1\}$ ir divi tuvākie kaimiņi, $N_i = \{i-1, i+1\}$, un mezglam pie robežām (diviem beigu galiem) ir katram viens kaimiņš, $N_1 = \{2\}$ un $N_m = \{m-1\}$. Kad mezgli atrodas regulārā taisnstūra režģī $S = \{(i, j) \mid 1 \leq i, j \leq n\}$ un atbilst pikseļiem $n \times n$ attēlā 2D plaknē, iekšējam mezglam (i, j) ir četri tuvākie kaimiņi $N_{i,j} = \{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\}$, mezglam pie attēla malas ir trīs un mezglam, kas atrodas attēla stūrī, ir divi.

Pāris $(S, N) = G$ izveido grafu parastajā nozīmē; S satur mezglus un N nosaka saites starp mezgliem, ņemot vērā kaimiņu attiecības. [7]

Izveido jēdzienu kliķe. Kliķe ir tāda apakškopa grafā, kas sastāv no mezgliem, ka starp visiem mezglu pāriem šajā kopā eksistē saite. Tas nozīmē, ka kliķe ir kopa, kas sastāv no mainīgajiem, kuri ir atkarīgi viens no otra.



3.2. att. Kliķes

Šis koncepts ir ilustrēts ar neorientētu grafu ar četrām virsotnēm, parādīts 3.2. att. Šim grafam ir piecas kliķes: $\{x_1, x_2\}$, $\{x_2, x_3\}$, $\{x_3, x_4\}$, $\{x_4, x_2\}$, $\{x_1, x_2, x_3\}$ un $\{x_2, x_3, x_4\}$. Kliķes $\{x_1, x_2, x_3\}$ un $\{x_2, x_3, x_4\}$ sauc par maksimālām, jo tās satur pārējās kliķes kā apakškopas. [3]

Kliķe var sastāvēt no mezglu pāriem, kuri ir kaimiņi, $c = \{i, i'\}$, trijnieka, kur mezgli ir kaimiņi, $c = \{i, i', i''\}$ un tā tālāk. Saimes no pāra un trijnieka mezgliem tiek apzīmētas ar C_2 , C_3 , kur

$$C_2 = \{\{i, i'\} | i' \in N_i, i \in S\};$$

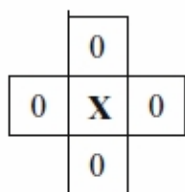
$$C_3 = \{\{i, i', i''\} | i, i', i'' \in S\};$$

kur i, i', i'' ir kaimiņi viens otram.

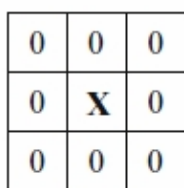
Mezgli kliķē ir sakārtoti, un $\{i, i'\}$ nav tā pati kliķe kā $\{i', i\}$ utt. Saime no visām kliķēm (S, N) ir

$$C = C_2 \cup C_3 \dots$$

Attiecīgi „...” norāda iespējamās kopas no lielākām kliķēm.



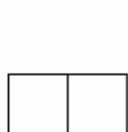
(a)



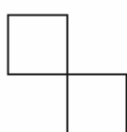
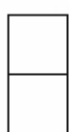
(b)

5	4	3	4	5
4	2	1	2	4
3	1	X	1	3
4	2	1	2	4
5	4	3	4	5

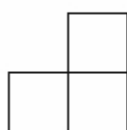
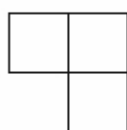
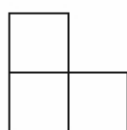
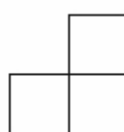
(c)



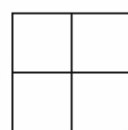
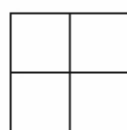
(d)



(e)



(f)



(h)

3.3. att. Kaimiņattiecības un kliķes režģī no regulāriem mezgliem

Regulārā režģī kliķes tips pārim (S, N) ir noteikts ar tā izmēru, formu un orientāciju. Figūras 3.3. att. (d) – (h) parāda kliķes tipus pirmās un otrās pakāpes kaimiņu sistēmām. Pirmās kārtas kaimiņu sistēmai (a) atbilst horizontālās un vertikālās pāra mezglu kliķes, kas parādītas piemērā (d). Kliķes tipi otrās kārtas kaimiņu sistēmai (b) satur ne tikai iepriekšējo (d), bet arī diagonālas pāra mezglu kliķes (e), trijnieka mezglu (f) un četrinieka mezglu (h) kliķes. Paaugstinoties kaimiņu sistēmas pakāpei, kliķu skaits arī strauji pieaug.

3.4.1.2. Markova gadījuma lauks

Apzīmē ar $F = \{F_1, \dots, F_m\}$ gadījuma lielumu saimi, kas definēta kopā S . Katrs gadījuma lielums F_i pieņem vērtību f_i kopā L . Saime F tiek saukta par gadījuma lauku. Lieto pierakstu $F_i = f_i$, lai norādītu gadījumu, ka F_i pieņem vērtību f_i un pierakstu $(F_1 = f_1, \dots, F_m = f_m)$, lai norādītu kopējo gadījumu. Vienkāršības labad kopējais notikums ir saīsināts kā $F = f$,

kur $f = \{f_1, \dots, f_m\}$ ir konfigurācija no F , kas atbilst lauka realizācijai. Diskrētai apzīmējumu kopai L varbūtība, ka gadījuma lielums F_i pieņem vērtību f_i ir apzīmēts $P(F_i = f_i)$, saīsināts $P(f_i)$, un kopējā varbūtība ir apzīmēta $P(F = f) = P(F_1 = f_1, \dots, F_m = f_m)$, saīsināta $P(f)$. Nepārtrauktai L ir dotas varbūtību blīvuma funkcijas, $p(F_i = f_i)$ un $p(F = f)$.

F tiek saukts par Markova gadījuma lauku kopā S attiecībā pret kaimiņu sistēmu N tad un tikai tad, ja izpildās divi nosacījumi:

1. $P(f) > 0 \quad \forall f \in F$ (pozitivitāte);
2. $P(f_i | f_{S-\{i\}}) = P(f_i | f_{N_i})$ (Markova nosacījums).

$S - \{i\}$ ir kopas starpība, $f_{S-\{i\}}$ norāda apzīmējumu kopu mezgliem $S - \{i\}$.

$f_{N_i} = \{f_{i'} | i' \in N_i\}$ norāda apzīmējumu kopu mezgliem, kas ir kaimiņi mezglam i . Pozitivitāte tiek pieņemta dažu tehnisku iemeslu dēļ un parasti var tikt izpildīta praksē. Piemēram, ja pozitivitātes nosacījums ir spēkā, tad katra gadījuma lauka kopējā varbūtība $P(f)$ ir unikāli noteikta ar tā lokālām nosacījuma varbūtībām. Markova nosacījums apraksta F lokālās īpašības. MRF modelī tieša mijiedarbība ir tikai kaimiņu apzīmējumiem.

MRF var piemist arī citas īpašības, piemēram, homogenitāte.

21. definīcija MRF sauc par homogēnu, ja varbūtība $P(f_i | N_i)$ ir neatkarīga no mezgla i atrašanās vietas režģī S . Tātad homogēnam MRF, ja $f_i = f_j$ un $f_{N_i} = f_{N_j}$, izpildīsies $P(f_i | N_i) = P(f_j | N_j)$ pat ja $i \neq j$.

Homogenitāte tiek pieņemta vairums MRF attēlu apstrādes modeļos matemātiskai un datoru aprēķinu vienkāršības dēļ.

Ir divas pieejas, kā specificēt MRF -ar nosacīto varbūtību $P(f_i | N_i)$ palīdzību un ar kopējo varbūtību $P(f)$.

3.4.1.3. Gibbsa gadījuma lauks

22. definīcija Kopa no gadījuma lielumiem F tiek saukta par Gibbsa gadījuma lauku (*angļu val.* Gibbs random field, *saīss.* GRF) režģī S attiecībā pret kaimiņu sistēmu N tad un tikai tad, ja tā konfigurācijai piemīt Gibbsa sadalījums. Gibbsa sadalījums pieņem formu:

$$P(f) = \frac{1}{Z} \cdot e^{-\frac{1}{T}U(f)}, \quad (3.3)$$

kur $Z = \sum_{f \in F} e^{-\frac{1}{T}U(f)}$ ir normalizējošā konstante. T ir konstante, kas tiek saukta par temperatūru,

kuru pieņem 1, ja tas nav citādi norādīts. $U(f)$ ir enerģijas funkcija. Enerģija

$$U(f) = \sum_{c \in C} V_c(f) \quad (3.4)$$

ir summa no visiem iespējamajiem kliķu C potenciāliem $V_c(f)$. Vērtība $V_c(f)$ ir atkarīga no lokālām kliķes c konfigurācijām.

GRF sauc par homogēnu, ja $V_c(f)$ ir neatkarīgs no kliķes c atrašanās vietas režģī S . To sauc par izotropu, ja V_c ir neatkarīgs no c orientācijas. Ir daudz vienkāršāk specificēt GRF sadalījumu, kas ir homogēns un izotropns nekā tādu, kuram nepiemīt šādas īpašības.

Lai aprēķinātu Gibbsa sadalījumu, ir nepieciešams noteikt normalizējošo funkciju Z , kas ir summa pār visām iespējamām konfigurācijām F . Tā kā pastāv kombinatorisks elementu skaits kopā F diskrētai L , aprēķins ir pārmērīgi augsts pat problēmām ar vidēju izmēru. Daudzas aproksimācijas metodes eksistē, lai atrisinātu šo problēmu.

Gibsa sadalījums reprezentē iedomātu fizikālu sistēmu, kuras zemākais enerģijas stāvoklis apraksta oriģinālo attēlu. $P(f)$ mēra varbūtību, ka iestājas konkrēta konfigurācija jeb „attēls” f . Vairāk iespējamās konfigurācijas ir tās, kurām ir lielāka varbūtība jeb zemākas enerģijas.

Diskrētām apzīmējuma piešķiršanas problēmām kliķes potenciāls $V_c(f)$ var tikt specificēts ar parametriem. Piemēram, gadījumā, ja $f_c = (f_i, f_{i'}, f_{i''})$ ir lokāla trijnieka $c = (i, i', i'')$ konfigurācija, f_c pieņem galīgu skaitu stāvokļu un tādēļ $V_c(f)$ pieņem galīgu skaitu vērtību. Nepārtrauktām apzīmējuma piešķiršanas problēmām, f_c var variēt bezgalīgi. Šajā gadījumā $V_c(f)$ ir nepārtraukta funkcija no f_c .

Dažkārt var būt ērti izteikt Gibbsa sadalījuma enerģiju kā summu no vairākiem lielumiem, katrs attiecināts konkrēta izmēra kliķei, t.i.,

$$U(f) = \sum_{\{i\} \in C_1} V_1(f_i) + \sum_{\{i,i'\} \in C_2} V_2(f_i, f_{i'}) + \sum_{\{i,i',i''\} \in C_3} V_3(f_i, f_{i'}, f_{i''}) + \dots \quad (3.5)$$

Izteiksme apzīmē homogēnu Gibbsa sadalījumu, jo V_1 , V_2 un V_3 ir neatkarīgi no i, i', i'' atrašanās vietas. Nehomogēniem Gibbsa sadalījumiem, kliķes funkcijas būtu jāraksta kā $V_1(i, f_i)$, $V_2(i, i', f_i, f_{i'})$ un tā tālāk.

Svarīgs speciāls gadījums ir, kad tiek ņemtas vērā tikai kliķes ar izmēru līdz divām. Šajā gadījumā enerģiju arī var pierakstīt sekojoši:

$$U(f) = \sum_{i \in S} V_1(f_i) + \sum_{i \in S} \sum_{i' \in N_i} V_2(f_i, f_{i'}). \quad (3.6)$$

Nosacītā varbūtība var tikt uzrakstīta kā

$$P(f_i | N_i) = \frac{e^{-\left[V_1(f_i) + \sum_{i' \in N_i} V_2(f_i, f_{i'}) \right]}}{\sum_{f_i \in L} e^{-\left[V_1(f_i) + \sum_{i' \in N_i} V_2(f_i, f_{i'}) \right]}}. \quad (3.7)$$

Ievietojot (3.6) izteiksmē (3.3), var uzrakstīt kopējo varbūtību kā reizinājumu

$$P(f) = Z^{-1} \prod_{i \in S} r_i(f_i) \prod_{i \in S} \prod_{i' \in N_i} r_{i,i'}(f_i, f_{i'}). \quad (3.8)$$

$$r_i(f_i) = e^{-\frac{1}{T} V_1(f_i)} \quad \text{un} \quad r_{i,i'}(f_i, f_{i'}) = e^{-\frac{1}{T} V_2(f_i, f_{i'})}.$$

3.4.1.4. Markova-Gibsa ekvivalence

Hammersley-Clifford teorēma pamato ekvivalenci starp Markova gadījuma lauku un Gibbsa gadījuma lauku. Teorēma apgalvo, ka F ir MRF režģī S attiecībā pret N tad un tikai tad, ja F ir GRF režģī S attiecībā pret N .

Pierādījums

$$GRF \Rightarrow MRF$$

$P(f)$ ir Gibbsa sadalījums režģī S attiecībā pret kaimiņu sistēmu N .

1. Pārbauda, vai ir spēkā pozitivitātes nosacījums. Ir acīmredzami, ka

$$P(f) > 0 \quad \forall f \in F, \text{ jo } P(f) = \frac{1}{Z} \cdot e^{-\frac{1}{T}U(f)}$$

2. Pierāda, ka GRF izpildās Markova nosacījums.

Aplūko nosacīto varbūtību

$$P(f_i | f_{S-\{i\}}) = \frac{P(f_i, f_{S-\{i\}})}{P(f_{S-\{i\}})} = \frac{P(f)}{\sum_{f'_i \in L} P(f')}. \quad (3.9)$$

kur $f' = \{f_1, \dots, f_{i-1}, f'_i, \dots, f_m\}$ ir jebkura konfigurācija, kas sakrīt ar f visos mezglos, izņemot i .

Izrakstot

$$P(f) = \frac{1}{Z} \cdot e^{-\sum_{c \in C} V_c(f)}, \quad (3.10)$$

dod

$$P(f_i | f_{S-\{i\}}) = \frac{e^{-\sum_{c \in C} V_c(f)}}{\sum_{f'_i} e^{-\sum_{c \in C} V_c(f')}} \cdot \quad (3.11)$$

Sadala C divās kopās - A un B . Kopa A sastāv no klikšēm, kas satur i , un B no klikšēm, kas nesatur i .

Tad (3.11) var uzrakstīt formā

$$P(f_i | f_{S-\{i\}}) = \frac{\left[e^{-\sum_{c \in A} V_c(f)} \right] \left[e^{-\sum_{c \in B} V_c(f)} \right]}{\sum_{f'_i} \left\{ \left[e^{-\sum_{c \in A} V_c(f')} \right] \left[e^{-\sum_{c \in B} V_c(f')} \right] \right\}} \cdot \quad (3.12)$$

Tāpēc ka $V_c(f) = V_c(f')$ katrai klikšei c , kas nesatur i , $e^{-\sum_{c \in B} V_c(f)}$ var izsvītrot gan no skaitītāja, gan saucēja.

Tāpēc šī varbūtība ir atkarīga tikai no to klikšu potenciāliem, kas satur i ,

$$P(f_i | f_{S-\{i\}}) = \frac{e^{-\sum_{c \in A} V_c(f)}}{\sum_{f_i'} e^{-\sum_{c \in A} V_c(f')}} \cdot \quad (3.13)$$

Varbūtība ir atkarīga tikai no apzīmējumiem pie i kaimiņiem. Tas pierāda, ka Gibbsa gadījuma lauks ir Markova gadījuma lauks.

Pierādījums, ka MRF ir GRF ir daudz sarežģītāks un netiks apskatīts. Daudzi *Hammersley-Clifford* teorēmas pierādījumu eksistē, piemēram, [1], [2], [6].

[12]

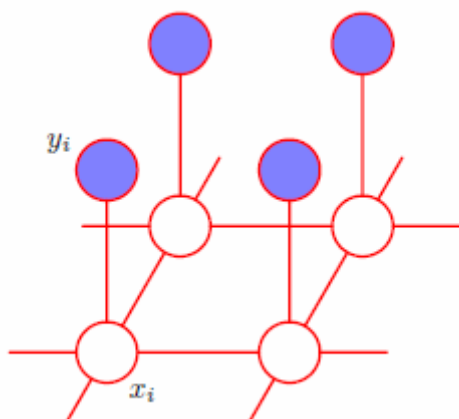
3.4.2. Enerģijas funkcijas loma

Enerģijas funkcija fizikālā sistēmā nosaka tās Gibbsa sadalījumu. Tā ir ērti un praktiski lietojama un sniedz dabiskāku mehānismu attēla īpašību izteikšanai nekā lokālās MRF pazīmes. Gibbsa gadījuma lauka un Markova gadījuma lauka ekvivalences dēļ šī funkcija arī nosaka MRF attēla modeli. Šī ekvivalence sniedz precīzu formulu kopējās varbūtības sadalījuma $P(F = f)$ aprēķināšanai enerģijas funkcijas veidā, kā arī spēcīgu mehānismu, lai modelētu telpisku nepārtrauktību un citas attēla pazīmes. [7]

3.4.3. Ising modelis

Visvairāk pazīstamā no šīm režģu sistēmām ir *Ising* modelis, kas palīdz izskaidrot feromagnētismu. Aplūko *Ising* metodes pielietojumam attēlu apstrādē.

Apskata trokšņainu attēlu, aprakstītu ar binārām pikseļa vērtībām $y_i \in \{-1, 1\}$, $i = 1, \dots, n$, kur indekss i norāda apskatīto pikseli. Pieņem, ka attiecīgais nepazīstamais attēls bez trokšņiem ir aprakstīts ar binārām pikseļa vērtībām $x_i \in \{-1, +1\}$, $i = 1, \dots, n$.



Neorientētais grafs reprezentē MRF beztrokšņu attēla iegūšanu no „trokšņaina” attēla, kur x_i norāda pikseļa i stāvokli nepazīstamajā beztrokšņu attēlā un y_i - atbilstošo pikseli i novērotajā trokšņainajā attēlā.

Pastāv saistība starp x_i un y_i . Ir arī zināms, ka savienotie pikseļi x_i un x_j ir atkarīgi. Šīs zināšanas var izmantot MRF modelī. Grafs satur divu veidu kliķes,

katra no tām satur divus mainīgos. Kliķes formā $\{x_i, y_i\}$ ir asociētas ar enerģijas funkciju, kas izsaka saistību starp šiem mainīgajiem. Izmanto ļoti vienkāršu enerģijas funkciju šīm kliķēm formā $-h \cdot x_i y_i$, kur h ir pozitīva konstante. Funkcijai piemīt vēlamais efekts - dodot zemāku enerģiju (piešķirot augstāku varbūtību), kad x_i un y_i ir tā paša zīme, un augstāku enerģiju, kad ir pretējas zīmes.

Paliekošās kliķes satur mainīgo pārus $\{x_i, x_j\}$, kur i un j ir indeksi no blakus esošiem pikseliem. Kā iepriekšējā gadījumā vēlas panākt, lai enerģija ir zemāka, kad pikseliem ir tā pati zīme, un augstāka, kad tiem ir pretējas zīmes. Un tā izvēlas enerģiju, kas ir dota $-b \cdot x_i x_j$, kur b ir pozitīva konstante.

Tāpēc, ka potenciālā funkcija ir patvaļīga, nenegatīva funkcija pār maksimālo kliķi, to var reizināt ar jebkuru nenegatīvu funkciju no maksimālās kliķes apakškopas vai ekvivalenti saskaitīt atbilstošās enerģijas. Šajā piemērā pieskaita papildus lielumu $h \cdot x_i$ katram pikselim beztrokšņu attēlā.

Šādam lielumam ir efekts, ka modelī kādai konkrētai zīmei tiek dota priekšroka pār citu.

Kopējā enerģijas funkcija modelim ir dota formā:

$$E(x, y) = h \cdot \sum_i x_i - b \sum_{\{i,j\}} x_i x_j - h \sum_i x_i y_i,$$

kura definē x un y sadalījuma funkciju sekojoši:

$$p(x, y) = \frac{1}{Z} \exp\{-E(x, y)\}.$$

Attēla atjaunošanai sameklē attēlu X , kuram ir augsta varbūtība (ideāli maksimālā varbūtība).

Ideja ir vispirms inicializēt mainīgos $\{x_i\}$, ko dara vienkārši nosakot $x_i = y_i$ visiem i . Tad paņem vienu mezglu x_j un aprēķina kopējo enerģiju iespējamām vērtībām $x_j = +1$ un $x_j = -1$, visus pārējos mezglu mainīgos turot fiksētus, un x_j novieto uz to stāvokli, kuram ir mazākā enerģija. Vai nu varbūtība paliks nemainīga, ja x_j netiek mainīts, vai to palielinās. Tikai viens mainīgais tiek mainīts, tas ir vienkāršs, lokāls aprēķins, kas var tikt izpildīts efektīvi. Tad atkārtoti atjaunošanu citam lokam un tā tālāk, līdz kāds piemērots apstāšanās kritērijs ir sasniegts. [3]

4. Nosacītais gadījuma lauks

Optimālo konfigurāciju, t.i., attēlu iegūst, meklējot varbūtības $P(f|d)$ maksimālo vērtību.

CRF modelē varbūtību $P(f|d)$ līdzīgi kā MRF. Apzīmējumu kopu f sauc par CRF pie dotiem datiem d , ja katrs f_i apmierina Markova nosacījumu (pieņemta arī pozitivitāte)

$$P(f_i|d, f_{S-\{i\}}) = P(f_i|d, f_{N_i}).$$

Balstoties uz Markova-Gibsa ekvivalenci, iegūst

$$P(f|d) = \frac{1}{Z} \exp\left(-\frac{1}{T} E(f|d)\right).$$

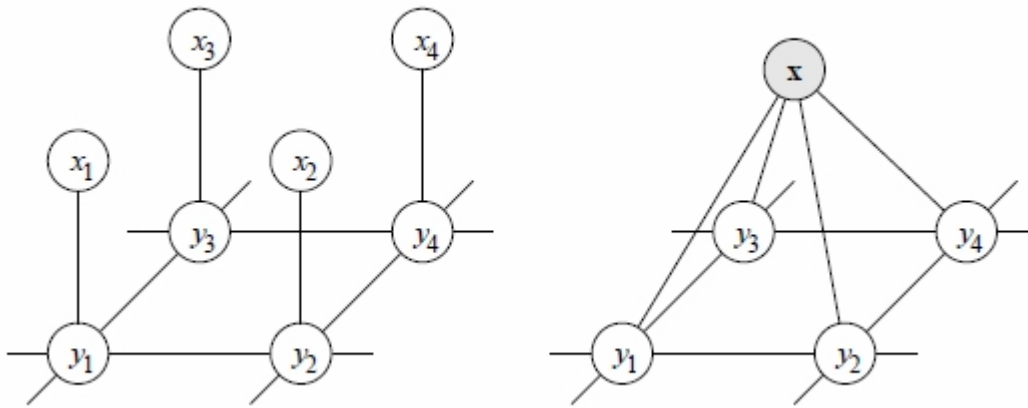
Z ir normalizējošā funkcija un $E(f|d)$ enerģijas funkcija. Ja vienīgie kliķes potenciāli, kas nav nulle ir tie, kas sastāv no viena un diviem mezgliem, varbūtība $P(f|d)$ ir formā

$$P(f|d) = \frac{1}{Z} \exp\left\{-\sum_{i \in S} V_1(f_i|d) - \sum_{i \in S} \sum_{i' \in N_i} V_2(f_i, f_{i'}|d)\right\}.$$

$-V_1$ un $-V_2$ tiek saukti par asociāciju un mijiedarbību potenciāliem. Vispārējā gadījumā šie potenciāli tiek aprēķināti kā lineāra kombinācija no novēroto datu īpašībām.

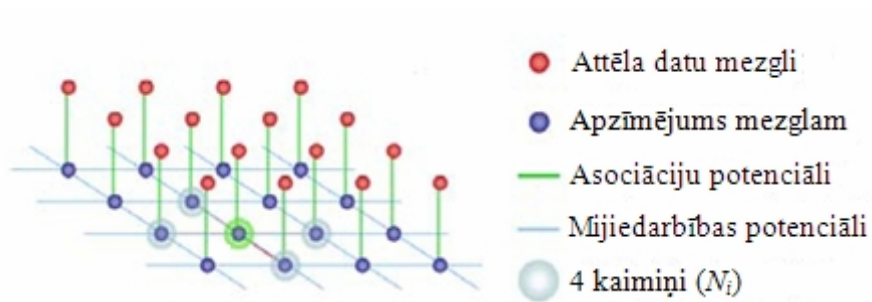
Ir divas būtiskas atšķirības starp CRF un MRF. Vispirms, CRF modelī asociāciju potenciāls mezglam i ir funkcija no visiem novērotajiem datiem d_1, \dots, d_n , kā arī no apzīmējuma f_i ; MRF modelī asociāciju potenciāls ir funkcija tikai no novērojumiem f_i un d_i . Otrkārt, MRF modelī mijiedarbības potenciāls katram mezglu pārim i un i' ir neatkarīgs no novērotajiem datiem; CRF tā ir arī funkcija no visiem d_1, \dots, d_n kā arī no apzīmējumiem f_i un $f_{i'}$. Atšķirībā no MRF, kur $d_{i'}$ var ietekmēt f_i ($i \neq i'$) netieši ar kaimiņu sistēmu, CRF modelī tas notiek tieši ar saikni starp $d_{i'}$ un f_i .

Uzskatāmi MRF un CRF atšķirību parāda 3.5. att. Modelis kreisajā pusē ir MRF, kur informācijas apmaiņa notiek tikai ar savienotajiem punktiem. Labajā pusē attēlots CRF, kur informācija tiek ņemta no visiem datu punktiem.



3.5. att. MRF un CRF

Iepriekš ieviesto CRF modeli var aplūkot arī kā MRF. Uzskatāmi to var redzēt 3.6. att.

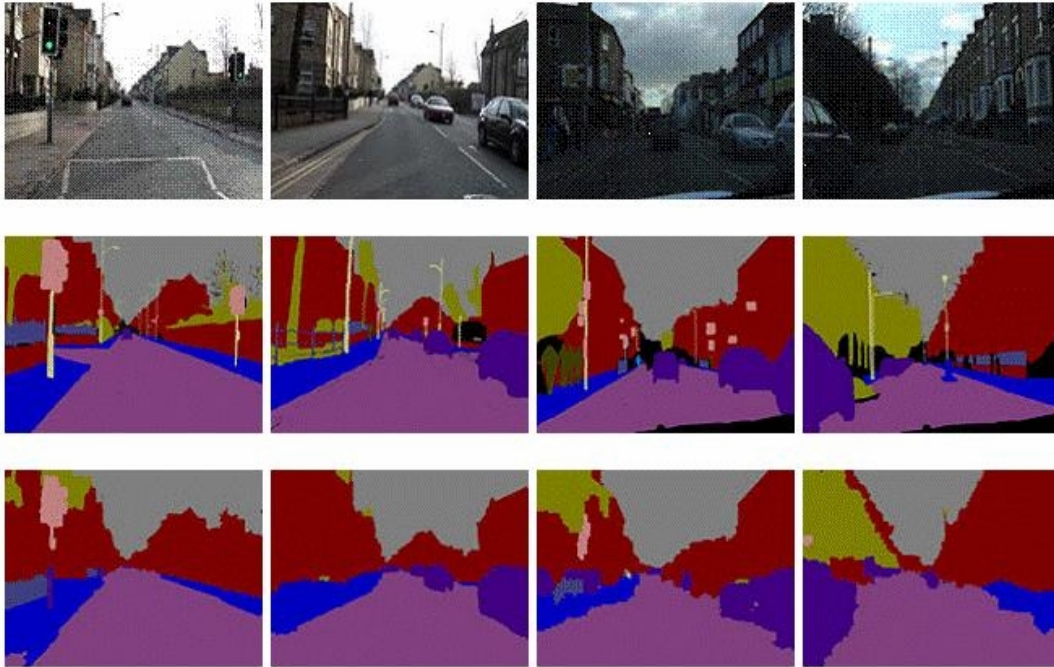


3.6. att. MRF modelis

[4]

5. Markova gadījuma lauka un nosacītā gadījuma lauka iegūtie rezultāti

Sasniegtie rezultāti ar CRF modeli ir līdzīgi kā MRF gadījumā. Attēlu segmentācijas problēma parasti tiek formulēta, izmantojot CRF modeli, jo tas sniedz iespēju ņemt vērā visus pieejamos datus d , lai noteiktu apzīmējumu f_i pikselim i .



3.7. att. Rezultāti ar MRF



3.8. att. Rezultāti ar CRF

NOBEIGUMS

Attēlu apstrāde ir interesanta un plaša pētījuma sfēra, kurai piemīt daudzas pielietojuma jomas. Viena no iespējām ir robotika, kuras nozīme mūsdienās arvien vairāk pieaug.

IURO projekta mērķis ir attīstīt un pielietot metodes un tehnoloģijas, lai padarītu iespējamu situāciju, kad roboti kustas un darbojas plaši apdzīvotās, nepazīstamās cilvēku koncentrētās vidēs un iegūst informāciju no cilvēkiem, lai sasniegtu konkrētu vietu pilsētā vai savstarpējas mijiedarbības mērķi.

Lai īstenotu šo rezultātu, ir nepieciešami labi sasniegumi arī attēlu segmentācijā. Pastāv vairāki matemātiski modeļi attēlu apstrādei. Šis darbs sniedz ieskatu par attēlu apstrādes algoritmiem, un aplūkotas tiek trīs metodes.

MRF un CRF modeļos var ņemt vērā pazīmes un ierobežojumus datos, kurus izvēlas. Līdzīgi nestrikta metrikā arī var ņemt vērā vairākus attāluma kritērijus, t.i., vairākas pazīmes. Tas rada priekšrocību salīdzinājumā ar citām metodēm. Nestrikto metrikas teorija ir saistošs koncepts, kā apstrādāt dažāda veida nenoteiktības attēlos. Realizējot šāda veida apstrādi, iegūst labākus rezultātus nekā ar klasiskajām metrikām. MRF un CRF paver iespēju apskatīt atkarību starp dotiem datiem pie dažādiem nosacījumiem.

Autors cer, ka attēlu apstrāde un pielietotās metodes ir kļuvušas labāk izprotamas un metožu potenciāli ir pamanāmi. Nākotnes vīzija ir izveidot robotu, kas patstāvīgi atradīs savu ceļu līdz iepriekš noteiktam objektam strauji mainīgā vidē, komunicējot ar garāmgājējiem.

IZMANTOTĀ LITERATŪRA UN AVOTI

1. Žurnāli

- [1] Besag, J.; *Spatial interaction and the statistical analysis of lattice systems (with discussions)*; Journal of the Royal Statistical Society, Series B 36; 1974; 192.–236. lpp.
- [2] Moussouris, J.; *Gibbs and Markov systems with constraints*; Journal of statistical physics 10; 1974; 11.–33. lpp.

2. Grāmatas

- [3] Bishop, C. M.; *Pattern Recognition and Machine Learning*; Springer; 2006.
- [4] Cipolla, R., Battiato, S., Farinella, G. M.; *Computer Vision Detection, Recognition and Reconstruction*; Springer-Verlag Berlin Heidelberg; 2010.
- [5] Gregori, V., Morillas, S., Sapena, A.; *Examples of fuzzy metrics and applications*; Elsevier B.V.; 2010.
- [6] Kindermann, R. and Snell, J. L.; *Markov Random Fields and Their Applications*. Providence, R.I.: American Mathematical Society; 1980.
- [7] Li, S.Z.; *Markov Random Field Modeling in Image Analysis*; Springer-Verlag London Limited; 2009.
- [8] Sapena, A. *A contribution to the study of fuzzy metric spaces, Applied General Topology* 2; 2001; 63.–76. lpp.
- [9] Šostaks, A., *L-kopas un L-vērtīgas struktūras*, Latvijas Universitāte; 2003.

3. Elektroniskie informācijas avoti

- [10] <http://cs.unimuenster.de/Studieren/Scripten/Lippe/wwwFuzzyScript/fsklassischfuzzy.html>
- [11] <http://www.gerhardmueller.de/docs/FuzzyLogic/node4.html>
- [12] <http://www.gerhardmueller.de/docs/FuzzyLogic/node5.html>

[13] <http://www.uni-weimar.de/medien/webis/teaching/lecturenotes/logics/unit-de-fuzzy-basics.pdf>

[14] http://theory.phy.umist.ac.uk/~judith/stat_therm/node87.html

[15] <http://www.vti-m.com/Website/Artikel/Fuzzy.pdf>

PIELIKUMI

1. boltzman

```
% Train model and denoise images

data=[];
for i=5:39
    A=strcat('Label_',int2str(i),'.mat');
    load (A);
    data=[data;double(V)];
end

testdata=[];
A=strcat('Label_',int2str(3),'.mat');
load (A);
testdata=[testdata;double(V)];

b=rand(size(testdata))<0.2;
noised=testdata;
r=rand(size(testdata));
noised(b)=r(b);

E = boltzman_backward(noised,height_small,width_small);
imshow(E);

%training
m2= rbmBB(data,80,'verbose',true);

%reconstruct the images by going up down then up again using learned model
up = rbmVtoH(m2, noised);
down = rbmHtoV(m2, up);

up = rbmVtoH(m2, noised);
down = rbmHtoV(m2, up);

%AGAIN
for i=1:50
    up = rbmVtoH(m2, down);
    down= rbmHtoV(m2, up);
end

E = boltzman_backward(down,height_small,width_small);
imshow(E);
```

2. rbmBB

```
function [model, errors] = rbmBB(X, numhid, varargin)
%Learn RBM with Bernoulli hidden and visible units
%This is not meant to be applied to image data
%code by Andrej Karpathy
%based on implementation of Kevin Swersky and Ruslan Salakhutdinov

%INPUTS:
%X          ... data. should be binary, or in [0,1] to be interpreted
%          ... as probabilities
```

```

%numhid          ... number of hidden layers

%additional inputs (specified as name value pairs or in struct)
%method          ... CD or SML
%eta             ... learning rate
%momentum        ... momentum for smoothness amd to prevent overfitting
%               ... NOTE: momentum is not recommended with SML
%maxepoch        ... # of epochs: each is a full pass through train data
%avglast         ... how many epochs before maxepoch to start averaging
%               ... before. Procedure suggested for faster convergence by
%               ... Kevin Swersky in his MSc thesis
%penalty         def 2e-4      ... weight decay factor
%batchsize       ... The number of training instances per batch
%verbose         ... For printing progress
%anneal          ... Flag. If set true, the penalty is annealed linearly
%               ... through epochs to 10% of its original value

%OUTPUTS:
%model.type      ... Type of RBM (i.e. type of its visible and hidden units)
%model.W         ... The weights of the connections
%model.b         ... The biases of the hidden layer
%model.c         ... The biases of the visible layer
%model.top       ... The activity of the top layer, to be used when training
%               ... DBN's
%errors          ... The errors in reconstruction at every epoch

%Process options
%if args are just passed through in calls they become cells
if (isstruct(varargin))
    args= prepareArgs(varargin{1});
else
    args= prepareArgs(varargin);
end
[ method          ...
  eta             ...
  momentum        ...
  maxepoch        ...
  avglast         ...
  penalty         ...
  batchsize       ...
  verbose         ...
  anneal          ...
  ] = process_options(args      , ...
    'method'      , 'CD'      , ...
    'eta'         , 0.1       , ...
    'momentum'    , 0.5       , ...
    'maxepoch'    , 45        , ...
    'avglast'     , 5         , ...
    'penalty'     , 2e-2      , ...
    'batchsize'   , 5         , ...
    'verbose'     , false     , ...
    'anneal'      , false);
avgstart = maxepoch - avglast;
oldpenalty= penalty;
[N,d]=size(X);

if (verbose)
    fprintf('Preprocessing data...\n');
end

%Create batches
numcases=N;

```

```

numdims=d;
numbatches= ceil(N/batchsize);
groups= repmat(1:numbatches, 1, batchsize);
groups= groups(1:N);
perm=randperm(N);
groups = groups(perm);
for i=1:numbatches
    batchdata{i}= X(groups==i,:);
end

%train RBM
W = 0.1*randn(numdims,numhid);
c = zeros(1,numdims);
b = zeros(1,numhid);
ph = zeros(numcases,numhid);
nh = zeros(numcases,numhid);
phstates = zeros(numcases,numhid);
nhstates = zeros(numcases,numhid);
negdata = zeros(numcases,numdims);
negdatastates = zeros(numcases,numdims);
Winc = zeros(numdims,numhid);
binc = zeros(1,numhid);
cinc = zeros(1,numdims);
Wavg = W;
bavg = b;
cavg = c;
t = 1;
errors=zeros(1,maxepoch);

for epoch = 1:maxepoch

    errsum=0;
    if (anneal)
        %apply linear weight penalty decay
        penalty= oldpenalty - 0.9*epoch/maxepoch*oldpenalty;
    end

    for batch = 1:numbatches
        [numcases numdims]=size(batchdata{batch});
        data = batchdata{batch};

        %go up
        ph = logistic(data*W + repmat(b,numcases,1));
        phstates = ph > rand(numcases,numhid);
        if (isequal(method,'SML'))
            if (epoch == 1 && batch == 1)
                nhstates = phstates;
            end
        elseif (isequal(method,'CD'))
            nhstates = phstates;
        end

        %go down
        negdata = logistic(nhstates*W' + repmat(c,numcases,1));
        negdatastates = negdata > rand(numcases,numdims);

        %go up one more time
        nh = logistic(negdatastates*W + repmat(b,numcases,1));
        nhstates = nh > rand(numcases,numhid);

        %update weights and biases
        dW = (data'*ph - negdatastates'*nh);

```

```

dc = sum(data) - sum(negdatastates);
db = sum(ph) - sum(nh);
Winc = momentum*Winc + eta*(dW/numcases - penalty*W);
binc = momentum*binc + eta*(db/numcases);
cinc = momentum*cinc + eta*(dc/numcases);
W = W + Winc;
b = b + binc;
c = c + cinc;

if (epoch > avgstart)
    %apply averaging
    Wavg = Wavg - (1/t)*(Wavg - W);
    cavg = cavg - (1/t)*(cavg - c);
    bavg = bavg - (1/t)*(bavg - b);
    t = t+1;
else
    Wavg = W;
    bavg = b;
    cavg = c;
end

%accumulate reconstruction error
err= sum(sum( (data-negdata).^2 ));
errsum = err + errsum;
end

errors(epoch)=errsum;
if (verbose)
    fprintf('Ended epoch %i/%i. Reconstruction error is %f\n', ...
        epoch, maxepoch, errsum);
end
end

model.type= 'BB';
model.top= logistic(X*Wavg + repmat(bavg,N,1));
model.W= Wavg;
model.b= bavg;
model.c= cavg;

```

3. examplecode

```

load mnist_classify;

% Train RBM for classification
%train rbm with 100 hidden units
m=rbmFit(data,100,labels,'verbose',true);
yhat=rbmPredict(m,testdata);

%print error
fprintf('Classification error using RBM with 100 hiddens is %f\n', ...
    sum(yhat~=testlabels)/length(yhat));

%visualize weights
figure(1)
visualize(m.W);
title('learned weights');

%visualize the mislabeled cases. Note the transpose. Visualize assumes DxN
%as is the case for weights
figure(2)
visualize(data(yhat~=testlabels,:));

```

```

title('classification mistakes for RBM with 100 hiddens');
drawnow;

%% Train model and denoise images
m2= rbmBB(data,100,'verbose',true);

%distort 100 images around by setting 95% to random noise
imgs=testdata(1:100,:);
b=rand(size(imgs))>0.2;
noised=imgs;
r=rand(size(imgs));
noised(b)=r(b);

%reconstruct the images by going up down then up again using learned model
up = rbmVtoH(m2, imgs);
down= rbmHtoV(m2, up);

%figure
z1=visualize(imgs');
z2=visualize(down');
figure(3)
imshow([z1 z2])
title('denoising 95% noise with RBM with 100 hidden units');
drawnow;

%AGAIN
for i=1:5
    up = rbmVtoH(m2, down);
    down= rbmHtoV(m2, up);
end
%figure
z1=visualize(imgs');
z2=visualize(down');

figure(4)
imshow([z1 z2])
title('denoising 95% noise with RBM and 2 passes');
drawnow;
%% Train a DBN
op.verbose=true;
models=dbnFit(data,[100 100],labels,op,op);
yhat2=dbnPredict(models,testdata);

%print error
fprintf('Classification error using DBN with 100-100 hiddens is %f\n', ...
        sum(yhat2~=testlabels)/length(yhat2));

%visualize weights
figure(4)
subplot(1,2,1)
visualize(models{1}.W);
title('learned weights on DBN layer 1');
subplot(1,2,2)
visualize(models{2}.W);
title('learned weights on DBN layer 2');

%visualize the mislabeled cases. Note the transpose. Visualize assumes DxN
%as is the case for weights
figure(5)
visualize(data(yhat2~=testlabels,:));
title('classification mistakes for DBN with 100-100 hiddens');

```

4. image_labeling

```
clear all

% Define the root folder for the images
HOMEIMAGES = '~/Labeled images'; % you can set here your default folder
HOMEANNOTATIONS = '~/Labeled images'; % you can set here your default
folder

% This line reads the entire database into a Matlab struct
database = LMdatabase(HOMEANNOTATIONS);

[D1,j1] = LMquery(database, 'object.name', 'building');
[D2,j2] = LMquery(database, 'object.name', 'car');
[D3,j3] = LMquery(database, 'object.name', 'people');
[D4,j4] = LMquery(database, 'object.name', 'sidewalk');
[D5,j5] = LMquery(database, 'object.name', 'street');
[D6,j6] = LMquery(database, 'object.name', 'barrier');
[D7,j7] = LMquery(database, 'object.name', 'sign');
[D8,j8] = LMquery(database, 'object.name', 'traffic light');
[D9,j9] = LMquery(database, 'object.name', 'bicycle');
[D10,j10] = LMquery(database, 'object.name', 'bush');
[D11,j11] = LMquery(database, 'object.name', 'grass');
[D12,j12] = LMquery(database, 'object.name', 'lamp');
[D13,j13] = LMquery(database, 'object.name', 'kiosk');
[D14,j14] = LMquery(database, 'object.name', 'column');
[D15,j15] = LMquery(database, 'object.name', 'dog');
[D16,j16] = LMquery(database, 'object.name', 'wall');
[D17,j17] = LMquery(database, 'object.name', 'sky');
[D18,j18] = LMquery(database, 'object.name', 'tree');

[height,width,dim]=size(imread('MG_7370.jpg'));

for im=1:40

image=zeros(height,width,dim);
imshow(image);

ind=find(j1==im);
if (ind>0)

% building - green
for i=1:max(size(D1(1,ind).annotation.object))
    hold on;
    fill(D1(1,ind).annotation.object(1,i).polygon.x,
D1(1,ind).annotation.object(1,i).polygon.y, 'g', 'LineStyle','none')
end;

    image=getframe;

end;

% car - red
ind=find(j2==im);

if (ind>0)

    for i=1:max(size(D2(1,ind).annotation.object))
        hold on;
```

```

        fill(D2(1,ind).annotation.object(1,i).polygon.x,
D2(1,ind).annotation.object(1,i).polygon.y, 'r', 'LineStyle','none');
    end;

    image=getframe;

end;

% tree - white
ind=find(j18==im);

if (ind>0)

    for i=1:max(size(D18(1,ind).annotation.object))
        hold on;
        fill(D18(1,ind).annotation.object(1,i).polygon.x,
D18(1,ind).annotation.object(1,i).polygon.y, 'w', 'LineStyle','none');
    end;

    image=getframe;

end;

% street - magenta
ind=find(j5==im);

if (ind>0)

    for i=1:max(size(D5(1,ind).annotation.object))
        hold on;
        fill(D5(1,ind).annotation.object(1,i).polygon.x,
D5(1,ind).annotation.object(1,i).polygon.y, 'm', 'LineStyle','none');
    end;

    image=getframe;

end;

% sidewalk - cyan
ind=find(j4==im);

if (ind>0)

    for i=1:max(size(D4(1,ind).annotation.object))
        hold on;
        fill(D4(1,ind).annotation.object(1,i).polygon.x,
D4(1,ind).annotation.object(1,i).polygon.y, 'c', 'LineStyle','none');
    end;

    image=getframe;

end;

% barrier - blue
ind=find(j6==im);

if (ind>0)

    for i=1:max(size(D6(1,ind).annotation.object))

```

```

        hold on;
        fill(D6(1,ind).annotation.object(1,i).polygon.x,
D6(1,ind).annotation.object(1,i).polygon.y, 'b', 'LineStyle','none');
    end;

    image=getframe;

end;

% bicycle - red
ind=find(j9==im);

if (ind>0)

    for i=1:max(size(D9(1,ind).annotation.object))
        hold on;
        fill(D9(1,ind).annotation.object(1,i).polygon.x,
D9(1,ind).annotation.object(1,i).polygon.y, 'r', 'LineStyle','none');
    end;

    image=getframe;

end;

% bush - white
ind=find(j10==im);

if (ind>0)

    for i=1:max(size(D10(1,ind).annotation.object))
        hold on;
        fill(D10(1,ind).annotation.object(1,i).polygon.x,
D10(1,ind).annotation.object(1,i).polygon.y, 'w', 'LineStyle','none');
    end;

    image=getframe;

end;

% grass - white
ind=find(j11==im);

if (ind>0)

    for i=1:max(size(D11(1,ind).annotation.object))
        hold on;
        fill(D11(1,ind).annotation.object(1,i).polygon.x,
D11(1,ind).annotation.object(1,i).polygon.y, 'w', 'LineStyle','none');
    end;

    image=getframe;

end;

% kiosk - green
ind=find(j13==im);

if (ind>0)

```

```

    for i=1:max(size(D13(1,ind).annotation.object))
        hold on;
        fill(D13(1,ind).annotation.object(1,i).polygon.x,
D13(1,ind).annotation.object(1,i).polygon.y, 'g', 'LineStyle','none');
    end;

    image=getframe;

end;

% column - blue
ind=find(j14==im);

if (ind>0)

    for i=1:max(size(D14(1,ind).annotation.object))
        hold on;
        fill(D14(1,ind).annotation.object(1,i).polygon.x,
D14(1,ind).annotation.object(1,i).polygon.y, 'b', 'LineStyle','none');
    end;

    image=getframe;

end;

% dog - blue
ind=find(j15==im);

if (ind>0)

    for i=1:max(size(D15(1,ind).annotation.object))
        hold on;
        fill(D15(1,ind).annotation.object(1,i).polygon.x,
D15(1,ind).annotation.object(1,i).polygon.y, 'b', 'LineStyle','none');
    end;

    image=getframe;

end;

% wall - blue
ind=find(j16==im);

if (ind>0)

    for i=1:max(size(D16(1,ind).annotation.object))
        hold on;
        fill(D16(1,ind).annotation.object(1,i).polygon.x,
D16(1,ind).annotation.object(1,i).polygon.y, 'b', 'LineStyle','none');
    end;

    image=getframe;

end;

% sky - dark green
ind=find(j17==im);

if (ind>0)

```

```

    for i=1:max(size(D17(1,ind).annotation.object))
        hold on;
        fill(D17(1,ind).annotation.object(1,i).polygon.x,
D17(1,ind).annotation.object(1,i).polygon.y, [0.3 0.5 0.3],
'LineStyle','none');
    end;

    image=getframe;

end;

% lamp - blue
ind=find(j12==im);

if (ind>0)

    for i=1:max(size(D12(1,ind).annotation.object))
        hold on;
        fill(D12(1,ind).annotation.object(1,i).polygon.x,
D12(1,ind).annotation.object(1,i).polygon.y, 'b', 'LineStyle','none');
    end;

    image=getframe;

end;

% traffic light - blue
ind=find(j8==im);

if (ind>0)

    for i=1:max(size(D8(1,ind).annotation.object))
        hold on;
        fill(D8(1,ind).annotation.object(1,i).polygon.x,
D8(1,ind).annotation.object(1,i).polygon.y, 'b', 'LineStyle','none');
    end;

    image=getframe;

end;

% sign - blue
ind=find(j7==im);

if (ind>0)

    for i=1:max(size(D7(1,ind).annotation.object))
        hold on;
        fill(D7(1,ind).annotation.object(1,i).polygon.x,
D7(1,ind).annotation.object(1,i).polygon.y, 'b', 'LineStyle','none');
    end;

    image=getframe;

end;

% people - yellow
ind=find(j3==im);

```

```

if (ind>0)

    for i=1:max(size(D3(1,ind).annotation.object))
        hold on;
        fill(D3(1,ind).annotation.object(1,i).polygon.x,
D3(1,ind).annotation.object(1,i).polygon.y, 'y', 'LineStyle','none');
    end;

    image=getframe;

end;

figure(2);
A=image.cdata(1:height,1:width,:);
imshow(A);
imwrite(A, strcat('Label_image',int2str(im),'.png'));

end;

```

5. boltzman_label

```

function [C,V] = boltzman_label(B,height,width)

C=zeros(height,width*8);

for i=1:height
    for j=1:width
        C(i,(j-1)*8+B(i,j))=1;
    end;
end;

V = vector(C,height,width);

```

6. boltzman_backward

```

function E = boltzman_backward(down,height_small,width_small)

lab_im=zeros(height_small,width_small*8);

for i=1:height_small
    lab_im(i,:)=down(width_small*8*(i-1)+1:width_small*i*8);
end;

D=zeros(height_small,width_small);

for i=1:height_small
    for j=1:width_small
        if size(find(max(lab_im(i,8*j-7:8*j))==lab_im(i,8*j-7:8*j)))==1
            D(i,j)=find(max(lab_im(i,8*j-7:8*j))==lab_im(i,8*j-7:8*j));
        else D(i,j)=9;
        end;
    end;
end;

E=uint8(zeros(height_small,width_small,3));

for i=1:height_small
    for j=1:width_small

```

```

    if D(i,j)==1
        E(i,j,1)=0 ; E(i,j,2)=255 ; E(i,j,3)=0;
    end;

    if D(i,j)==2;
        E(i,j,1)=255 ; E(i,j,2)=0 ; E(i,j,3)=0;
    end;

    if D(i,j)==3;
        E(i,j,1)=255 ; E(i,j,2)=255 ; E(i,j,3)=255;
    end;

    if D(i,j)==4;
        E(i,j,1)=255 ; E(i,j,2)=0 ; E(i,j,3)=255;
    end;

    if D(i,j)==5;
        E(i,j,1)=0 ; E(i,j,2)=255 ; E(i,j,3)=255;

    end;

    if D(i,j)==6;
        E(i,j,1)=0 ; E(i,j,2)=0 ; E(i,j,3)=255;
    end;

    if D(i,j)==7;
        E(i,j,1)=76 ; E(i,j,2)=127 ; E(i,j,3)=76;
    end;

    if D(i,j)==8;
        E(i,j,1)=255 ; E(i,j,2)=255 ; E(i,j,3)=0;
    end;

    if D(i,j)==9;
        E(i,j,1)=0 ; E(i,j,2)=0 ; E(i,j,3)=0;
    end;

end;
end;

```

7. adjacency

```

adj = sparse(nNodes,nNodes);

ind = 1:nNodes;
%wir schliessen den rechten Knoten fuer die letzte Kolonne aus, weil sie
%keinen Knoten rechts von sich hat
exclude = sub2ind([nRows nCols],1:nRows, repmat(nCols,[1 nRows]));
ind = setdiff(ind,exclude);
%wir schliessen die letzte Row aus, weil sie keinen Knoten unten hat
exclude = sub2ind([nRows nCols], repmat(nRows,[1 nCols]),1:nCols);
ind = setdiff(ind,exclude);
adj(sub2ind([nNodes nNodes],ind,ind+nRows+1)) = 1;

ind = 1:nNodes;
%wir schliessen den rechten Knoten fuer die letzte Kolonne aus, weil sie
%keinen Knoten rechts von sich hat
exclude = sub2ind([nRows nCols],1:nRows, repmat(nCols,[1 nRows]));
ind = setdiff(ind,exclude);

```

```

%wir schliessen die erste Row aus, weil sie keinen Knoten oben hat
exclude = sub2ind([nRows nCols], repmat(1,[1 nCols]),1:nCols);
ind = setdiff(ind,exclude);
adj(sub2ind([nNodes nNodes],ind,ind+nRows-1)) = 1;

% Add Down Edges
ind = 1:nNodes;
exclude = sub2ind([nRows nCols], repmat(nRows,[1 nCols]),1:nCols); % No Down
edge for last row
ind = setdiff(ind,exclude);
adj(sub2ind([nNodes nNodes],ind,ind+1)) = 1;

% Add Right Edges
ind = 1:nNodes;
exclude = sub2ind([nRows nCols],1:nRows, repmat(nCols,[1 nRows])); % No
right edge for last column
ind = setdiff(ind,exclude);
adj(sub2ind([nNodes nNodes],ind,ind+nRows)) = 1;

adj = adj+adj';

```

8. CRF_Training

```

clc
clear all
close all
tic

%% Make Xnode and y

Level           =4;                %Level of the image
nInstances       =1;                %Number of training Images
Evaluation       ='1';              %The Image to evaluate when we
want just to make the Xnode for a specific Image
%nInstances must be 1
nStates          =8;                %Number of States of the
Labeling Image
[A,B,C,D]
=Features(strcat('Real_Image_',Evaluation, '.png'),strcat('Label_Image_',Eva
luation, '.png'),Level);
[nNodes,features] =size(C);
Xnode=ones(nInstances,features+1,nNodes);

for j=1:nInstances
    for i=1:features
        Xnode(j,i+1,:)=C(:,i);
    end

    for i=1:length(D)
        if (D(1,i)==0)
            D(1,i)=1;
        elseif (D(1,i)==1)
            D(1,i)=2;
        elseif (D(1,i)==2)
            D(1,i)=3;
        elseif (D(1,i)==3)
            D(1,i)=4;
        elseif (D(1,i)==4)
            D(1,i)=5;
        end
    end
end

```

```

elseif (D(1,i)==5)
    D(1,i)=6;
elseif (D(1,i)==6)
    D(1,i)=7;
elseif (D(1,i)==7)
    D(1,i)=8;
else
    D(1,i)=1;
end
end

if(j==1)
    y=D;
else
    y=[y;D];
end

if (nInstances>1 & j<nInstances)
    [A,B,C,D]
=Features(strcat('Real_Image_',int2str(j+1),'.png'),strcat('Label_Image_',i
nt2str(j+1),'.png'),Level);
end

end

D=y;

[nRows,nCols,d] = size(A);

%% Make edgeStruct

adj = sparse(nNodes,nNodes);

% Add Down Edges
ind = 1:nNodes;
exclude = sub2ind([nRows nCols], repmat(nRows,[1 nCols]),1:nCols); % No Down
edge for last row
ind = setdiff(ind,exclude);
adj(sub2ind([nNodes nNodes],ind,ind+1)) = 1;

% Add Right Edges
ind = 1:nNodes;
exclude = sub2ind([nRows nCols],1:nRows, repmat(nCols,[1 nRows])); % No
right edge for last column
ind = setdiff(ind,exclude);
adj(sub2ind([nNodes nNodes],ind,ind+nRows)) = 1;

% Add Up/Left Edges
adj = adj+adj';
edgeStruct = UGM_makeEdgeStruct(adj,nStates);

%% Normalize Xnode and Make Xedge, infoStruct, initialize weights

tied = 1;
ising = 1;

Xnode(:,[2:features+1],:)=UGM_standardizeCols(Xnode(:,[2:features+1],:),1);

% Make Xedge

```

```

%sharedFeatures = [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
sharedFeatures = [1 0 0 ];
Xedge = UGM_makeEdgeFeatures(Xnode,edgeStruct.edgeEnds,sharedFeatures);
infoStruct = UGM_makeCRFInfoStruct(Xnode,Xedge,edgeStruct,ising,tied);

%% Train with Pseudo-likelihood
%*****
% % % Initialize weights
% [w,v] = UGM_initWeights(infoStruct);
% inferFunc = @UGM_Infer_MeanField;
%                                     funObj                               =
@(wv)UGM_CRFLoss(wv,Xnode,Xedge,D,edgeStruct,infoStruct,inferFunc);
% [wv] = minFunc(funObj,[w(:);v(:)]);
% [w,v] = UGM_splitWeights(wv,infoStruct);

% save ('Weights.mat','w','v');

%% Train with Descent Gradient

maxIter = 1000;
stepSize = 1e-4;
[w,v] = UGM_initWeights(infoStruct);
wv = [w(infoStruct.wLinInd);v(infoStruct.vLinInd)];
for iter = 1:maxIter
    i = ceil(rand*nInstances);
    funObj                                     =
@(wv)UGM_CRFLoss(wv,Xnode(i,:,:),Xedge(i,:,:),y(i,:),edgeStruct,infoStruct,
@UGM_Infer_LBP);
    [f,g] = funObj(wv);

    %fprintf('Iter = %d of %d (fsub = %f)\n',iter,maxIter,f);

    wv = wv - stepSize*g;
end

[w,v] = UGM_splitWeights(wv,infoStruct);

save ('Weights.mat','w','v');

clc
toc

```

9. label_images

```

function [B,C,V] = label_images(A,height,width)

B=zeros(height,width);

for i=1:height
    for j=1:width

        if A(i,j,1)==0 && A(i,j,2)==255 && A(i,j,3)==0
            B(i,j)=1;

```

```

end;

if A(i,j,1)==255 && A(i,j,2)==0 && A(i,j,3)==0
B(i,j)=2;
end;

if A(i,j,1)==255 && A(i,j,2)==255 && A(i,j,3)==255
B(i,j)=3;
end;

if A(i,j,1)==255 && A(i,j,2)==0 && A(i,j,3)==255
B(i,j)=4;
end;

if A(i,j,1)==0 && A(i,j,2)==255 && A(i,j,3)==255
B(i,j)=5;
end;

if A(i,j,1)==0 && A(i,j,2)==0 && A(i,j,3)==255
B(i,j)=6;
end;

if A(i,j,1)==0 && A(i,j,2)==0 && A(i,j,3)==0
B(i,j)=6;
end;

if A(i,j,1)==76 && A(i,j,2)==127 && A(i,j,3)==76
B(i,j)=7;
end;

if A(i,j,1)==255 && A(i,j,2)==255 && A(i,j,3)==0
B(i,j)=8;
end;

end;
end;

[C,V] = boltzman_label(B,height,width);

```

10. feature

```

function [LQ,MQ,UQ] = feature(A)

A=rgb2gray(A);
[nRows,nCols] = size(A);
LQ=zeros(nRows,nCols);
MQ=zeros(nRows,nCols);
UQ=zeros(nRows,nCols);
for i=1:nRows-2
    for j=1:nCols-2
        X = [A(i,j) A(i+1,j) A(i+2,j) A(i,j+1) A(i+1,j+1) A(i+2,j+1)
A(i,j+2) A(i+1,j+2) A(i+2,j+2)];
        [LQ(i+1,j+1),MQ(i+1,j+1),UQ(i+1,j+1)]=quartile(X);
    end;
end;

```

11. quartile

```
function [Q1,Q2,Q3] = quartile(x)

y = sort(x);

% compute 25th percentile (first quartile)
Q1 = median(y(find(y<=median(y)))));

Q2 = median(y);

% compute 75th percentile (third quartile)

Q3 = median(y(find(y>=median(y)))));
```

Diplomdarbs „Nestrikas un varbūtiskas metodes attēlu apstrādei: teorētiskie aspekti un lietojumi robotu vadībai” izstrādāts LU Fizikas un matemātikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: Jana Ivanova

(paraksts)

(datums)

Rekomendēju darbu aizstāvēšanai.

Vadītājs: prof. Aleksandrs Šostaks

(paraksts)

(datums)

Recenzente: docents Ingrīda Uljane

Darbs iesniegts Matemātikas nodaļā

(datums)

(darbu pieņēma)

Darbs aizstāvēts valsts pārbaudījuma komisijas sēdē

_____ prot. Nr. _____, vērtējums _____

(datums)

Komisijas sekretāre: asoc. prof. Dr. math. Inese Bula

(paraksts)