

# UZ Povzetek

Yannick Kuhar

February 24, 2019

# 1 Image formation

## 1.1 Film image

Če postavimo objekt pred film ne dobimo dobre slike. Dodamo bariero, ki blokira večino žarkov zmanjša zameglitev. Odprtina, ki prepušča žarke se imenuje **aperture**(*ang.*).

Učinki velikosti **apertureja**:

- **prevelik**, prepusti preveč svetlobe kar povzroči zameglitev slike
- **premajhen**, prepusti premalo svetlobe kar povzroči zameglitev slike

V splošnem sta oba ekstrema slaba. Težvo povzroca na sploh majhno stevilo žarkov, ki zadanejo film in nastane temna slika.

To odpravimo z dajajo leče med objekt in film. Leča fokusira žarke na film. Točke na nekaterih razdalijah ostanejo fokusirane na nekatih pa zamegljene. Pri uporabi tanke leče, točke na različnih globinah fokusirajo na različnih globinah slikovne ravnine(*ang.* **image plane**).

**Depth of field** zardalija med **slikovnim ravninam** med katerimi je učinek zameglitve dovolj majhen. Majhni **aperture** poveča **depth of field**.

**Field of view(FOV)** ( $2 \times \varphi$ ) je kotna mera prostora, ki ga zazna kamera.

$$\varphi = \tan^{-1}\left(\frac{d}{2f}\right)$$

**Field of view**:

- majhen **focal length**( $f$ ) prinese širokokotno sliko
- visoko **focal length**( $f$ ) prinese teleskopsko sliko

**Aberacije**:

- **Chromatic aberration**, različne valovne dolžine se lomijo pod različnimi koti in fokusirajo pod različnimi razdalijami
- **Spherical aberration**, sferične leče ne fokusirajo svetlobo perfektno, žarki blizje robu se fokusirajo bližje kot žarki pri centru

**Vignetting** se zgodi, ko imamo več kot eno lečo v kameri. Fokusirani žarki iz prve (proti centru) leče se lomijo preko druge. Ker so žarki iz prve usmerjeni proti centru se namo prseslikajo preko druge (žarki pod kotom se filtrirajo). To povzroči temne kote.

Pogosta nepravilnost je tudi t.i. **Radial distortion** (*ang.*), ki se zgodi zaradi neprefektnosti leč.

## 1.2 Digital image

Namesto filma uporabimo matriko (tabelo) senzorjev. Diskretiziramo sliko v piksele in kvantiziramo svetlobo v intenzitetne nivoje.

## 1.3 CCD vs CMOS

- v obeh fotoni povzročijo naboj na senzorjih
- **CCD** prebere naboj po principu **FIFO** in digitalizira
- **CMOS** digitalizira vsak piksel posebj
- **CCD** slike so boljše kvalitete
- **CMOS** je cenejši za izdelavo

Zaradi osvetljenosti, ki je določena s strani zelene barve, uporabimo dvakrat toliko zelenih senzorjev kot modrih in rdečih. Človeški vid je bolj občutljiv na spremembe intenzitete kot na barvne spremembe (**Bayer sensor**).

## 2 Image processing 1

Binarne slike imajo samo dve možnosti intenzitete:

- **ospredje**(1)
- **ozadje**(0)

## 3 Image thresholding

Spremenimo sliko v binarno masko. Spoznali smo tri načine:

- **single threshold**  $F_T[i, j] = \begin{cases} 1, & \text{if } F[i, j] \geq T \\ 0, & \text{sicer} \end{cases}$
- **two thresholds**  $F_T[i, j] = \begin{cases} 1, & \text{if } T_1 \leq F[i, j] \leq T_2 \\ 0, & \text{sicer} \end{cases}$
- **apply a classifier**  $F_T[i, j] = \begin{cases} 1, & \text{if } F[i, j] \in Z \\ 0, & \text{sicer} \end{cases}$

Binarne slike lahko opišemo z **bimodalnim histogramom** (ima le dva stolpca in predstavlja porazdelitev pikslov).

### 3.1 Globalna binarizacija

Najdi takšen  $T$ , ki minimizira varianco med razredi, katere  $T$  ločuje.

$$\sigma_{within}^2(T) = n_1(T)\sigma_1^2 + n_2(T)\sigma_2^2$$

$$n_1(T) = |\{I_{(x,y)} < T\}|, \quad n_2(T) = |\{I_{(x,y)} \geq T\}|$$

Oziroma drugače maksimiziramo vmesen razred.

$$\sigma_{between}^2(T) = \sigma^2 - \sigma_{within}^2(T) = n_1(T)n_2(T)[\mu_1(T) - \mu_2(T)]^2$$

### 3.2 Otsujev algoritem

---

**Algorithm 1** Otsu's algorithm

---

```
1: procedure OTSU(I)
2:    $nbins \leftarrow 256$ 
3:    $counts \leftarrow myhist(I, nbins)$  % Get histohram
4:    $p \leftarrow counts / sum(counts)$  % Normalize the histohram
5:    $\sigma \leftarrow zeros(nbins, 1)$ 
6:   for  $t = 1 : nbins$  do
7:      $qlow \leftarrow sum(p(1 : t))$ 
8:      $qhigh \leftarrow sum(p(t + 1 : end))$ 
9:      $\mu_L \leftarrow sum(p(1 : t) .* (1 : t)) / qlow$ 
10:     $\mu_H \leftarrow sum(p(t + 1 : end) .* (t + 1 : nbins)) ./ qhigh$ 
11:     $\sigma(t) \leftarrow qlow * qhigh * (\mu_L - \mu_H)^2$ 
12:   $[\sim, threshold] \leftarrow max(\sigma(:))$ 
  return threshold
```

---

#### Neformalno

Za vsak **threshold T** razdeli sliko na dva razreda. Izračunaj  $\sigma$  vmesnega razreda in jo shrani. Vrni tak  $\sigma$ , ki maksimizira razpršenost vmesnega razreda.

### 3.3 Lokalna binarizacija

Ocenimo lokalni **threshold T** na oknu velikosti **W**:

$$T_W = \mu_W + k \times \sigma_W$$

In to storimo na vsaki regiji. Postopek se imenuje **Niblack's algorithm**.

## 4 Čiščenje slik

**Thresholding** ne odstrani šuma z slik, zato potrebujemo dodatno procesiranje. To storimo z morfološkimi operatorji.

### 4.1 Morfologija

#### Structuring element

Je vsebina premikajočega okna s katerim se pomikamo po sliki. Na primer:

$$SE = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

#### Fitting & Hitting

- **Fit**, SE in regija imata vse "1" istoležne
- **Hit**, SE in regija imata vsaj eno istoležno "1"

#### Erosion

Funkcijo slike **f** in **structuring element s** označimo z  $g = f \ominus s$  oziroma:

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

#### Dilation

Funkcijo slike **f** in **structuring element s** označimo z  $g = f \oplus s$  oziroma:

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

#### Combined operations

- **Opening**, erosion in dilation, odstrani majhne objekte in ohrani grobe oblike
- **Closing**, dilation in erosion, napolni luknje in ohrani originalno obliko

## 5 Označevanje regij

Procesiramo sliko od leve proti desni, od zgoraj dol:

1. Če je vrednost trenutnega piksla je 1
  - (a) če je zgornji ali levi sosed 1 kopiraj oznako
  - (b) če sta oba zgornji in levi 1 in imata enako oznako jo kopiraj
  - (c) če imata različne oznake kopiraj levo in posidobi tabelo ekvivalentnih oznak
  - (d) sicer uporabi novo oznako
2. ponovno označi z najmanjšim številom ekvivalentnih oznak

## 6 Region descriptors

Želimo takšen **descriptor**, ki slika dve sliki z podobnim objektom blizu in z različnim objektom daleč.

## 7 Image processing 2

### 7.1 Svetloba

Svetloba je elektromagnetska radiacija sestavljena iz več frekvenc. Lastnosti svetlobe so opisane v njenem spektru. Ljudje prepoznavamo svetlobo s pomočjo dveh tipov celic **cones** za barve(**R**, **G**, **B**) in **rods** za intenziteto.

### 7.2 Additive mixture model

Je klasičen **RGB** sistem. Kjer črni barvi dodajamo rdečo, modro in zeleno.

### 7.3 Subtractive models

Je **CMYK** model(**cyan**, **magenta**, **yellow**, **in key**).

## 7.4 Color spaces

Omogoča reprodukcijo barv. Vsak prostor ima primarne barve in vsaka nova barva je utežena vsota primarnih barv. Razdalije med barvbami so evklidske.

## 7.5 HSV colorspace

Hue (barvnost), saturation (nasičenje), value (intenziteta), je nelinearen barvni prostor.

# 8 Podobnost barv z histogrami

Histogram je stolpični diagram, ki meri nivoje intenzitete npr:

$h(i)$  = št. pikslov v sliki z intenziteto  $i$ .

$h(i) = \text{card}\{(u, v) \mid I(u, v) = i\}$  (**card** je funkcija, ki oceni vrednost)

Lahko imamo tudi barvne histograme:

$h(r, g, b)$  = št. pikslov v sliki z barvo (r, g, b).

## 8.1 Normalizacija intenzitete

Intenziteta je vsebovana v vsakem barvnem kanalu. Množenje barve z skaljarjem spremeni intenziteto ampak ne barve. To pomeni, da lahko normaliziramo intenziteto ( $\mathbf{I} = \mathbf{R} + \mathbf{B} + \mathbf{G}$ ) barve. Kromatska reprezentacija:

$$r = \frac{R}{R+B+G}, b = \frac{B}{R+B+G}, g = \frac{G}{R+B+G}, r + b + g = 1$$

Zdaj lahko uporabimo 2D prostor(rg). Lahko pa tudi indirektno primerjamo slike tako, da primerjamo njihove histograme(Evklidska razdalija, Hellingerjeva razdalija, ...).



## 9 Filtriranje

Glavna naloga je reduciranje šuma v sliki.

### 9.1 Tipi šuma

- **Sol in poper**, naključne bele in črne točke na sliki
- **Impulzivni šum** naključne bele točke na sliki
- **Gaussov šum** variacija intenzitete je vzorčena normalno

### 9.2 Poskusimo odstraniti piksle

Piksli so podobni kot sosednji. Predpostavimo, da je šum

**I.I.D(independent, identically distributed)**.

Sedaj lahko izračunamo boljšo oceno intenzitete piksla tako, da ga zamenjamo z povprečno intenziteto soseske pikslov. Predpostavimo, da imajo vsi piksli enako utež in soseska velikosti  $2k+1$ :

$$G[i, j] = \frac{1}{(2k+1)^2} \sum_{u=-k}^k \sum_{v=-k}^k F[i+u, j+v]$$

Oziroma bolj splošno, če dodamo uteži, ki niso enake:

$$G[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k H[u, v] F[i+u, j+v]$$

Kjer so  $H[u, v]$  neenake uteži.

Temu se reče **cross-correlation** oz. **correlation filtering** ( $G = H \otimes F$ ),  $H$  je okno in  $F$  je slika.

### 9.3 Konvolucija kot korelacija

Izračunamo konvolucijo z **cross-correlation**. Obrnemo filter v obe dimenzijah (horizontalno in vertikalno) in nato izvedemo **cross-correlation** in to označimo  $G = H \star F$  oziroma:

$$G[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k H[u, v] F[i-u, j-v]$$

## 9.4 Convolution vs. Correlation

- **Correlation**  $G[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k H[u, v] F[i + u, j + v]$ ,  $G = H \otimes F$
- **Convolution**  $G[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k H[u, v] F[i - u, j - v]$ ,  $G = H \star F$

Za simetričen filter  $H[-u, -v] = H[u, v]$  velja **correlation**  $\equiv$  **convolution**

## 9.5 Linearni shift-invariant sistem

**Shift-invariant** pomeni enako obnašanje ne glede na pozicijo oz. izhod je odvisen le od lokalnega vzorca ne pa od koordinat.

## 9.6 Lastnosti konvolucije

- linearnost, shift-invariant
- komutativnost:  $f * g = g * f$
- asociativnost:  $(f * g) * h = f * (g * h)$ , posledica tega je da uporaba večfiltrov je ekvivalentna uporabi enega  $((f * b_1) * b_2) * b_3 = f * (b_1 * b_2 * b_3)$
- identiteta: enotni impulz  $e = [\dots, 0, 0, 1, 0, 0, \dots]$
- odvod:  $\frac{\partial}{\partial x}(f * g) = (\frac{\partial}{\partial x}g) * f = (\frac{\partial}{\partial x}f) * g$

## 9.7 Gaussovo glajenje

Uporabimo **Gaussian kernel**:

$$G_{\sigma} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Piksli pri centru imajo večjo utež. Filter je simetričen glede na rotacijo (transponiranje ne spremeni ničesar). Varianca  $\sigma^2$  določi mero glajenja.

## 9.8 Učinkovita implementacija

Izvedemo konvolucijo nad vsako vrstico posebej z 1D kernelom:

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-x^2/(2\sigma^2))$$

Nato izvedemo 1D konvolucijo nad vsakim stolpcem:

$$g(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-y^2/(2\sigma^2))$$

Ta ločitev je mogoča zaradi linearnosti, asociativnosti in komutativnosti konvolucije oz:

$$g_x * (g_y * I) = (g_x * g_y) * I$$

## 9.9 Robni pogoji filtriranja

Kernel presega robove slike. Kaj storimo na robovih? Uporabimo štiri metode:

- **Crop**(black)
- **Bend** image
- **Replicate** edges
- **Mirror** image

## 9.10 Konvolucija in spekter

Konvolucija dveh funkcij je ekvivalentna produktu njunih Fourierjevih transformacij(spekter):

$$F(f * g) = F(f) \odot F(g)$$

Konvolucija manipulira slikovni spekter bodisi poudari bodisi zmanjša frekvenčni pass v sliki.

### 9.11 Ponovitev Fourierjeve transformacije

Signal je definiran ko vsota sinusov in cosinusov različnih frekvenc:

$$f(x) = \sum_n a_n \cos(nx) + b_n \sin(nx)$$

Šum je v resnici prisotnost visokih frekvenc. Odstranimo ga tako, da izvedemo t.i. **low-band pass** filter. Uporabimo lahko tudi t.i. **spatial box** filter vendar ta ima stranske učinke. Najbolj primeren **low-band pass** filter je **Gaussov**.

### 9.12 Sharpening filter

$$I * \left( \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right) = I_s$$

$I_s$  je izostrena slika. Sharpening filter poudari razlike z lokalnim povprečenjem.

### 9.13 Median filter

Je nelinearen filter. Osnovna ideja je zamenjati intenziteto piksla z mediano drsečega okna. Pomembni lastnosti sta:

- Ne dodaja novih sivih nivojev v sliko
- odstrani impulzne šume npr. **salt & pepper**

## 9.14 Filtering as template matching

Z metodami filtriranja lahko tudi iščemo lokacijo objekta na sliki. Template je lahko večji ali manjši kot slika zato moramo izvesti korelacijo za različne velikosti slike. Vendar namesto templatea skaliramo vhod. Učinkovit način spreminjanja velikosti slik je t.i. **Slikovna piramida**.

### 9.14.1 Kako zmanjšamo sliko?

Naivni pristop je odstranitev vsakega drugega piksla. Učinku pravimo **Aliasing**. Pristo povzroča težave saj spremeni objekte na sliki.

### 9.14.2 Prekrivanje (sampling and aliasing)

Zvezni signal diskretiziramo, izvedemo Fourierjevo transformacijo. Iz magnitude spektra izberemo eno kopijo in jo pretvorimo nazaj z inverzno Fourierjevo transformacijo (slika na Image processing 2 slide 68). Če želimo rekonstruirati vse frekvence do  $f$  potem moramo vzorčiti signale z frekvenco enako vsaj  $2f$ . Na neki točki se začnejo ponavljajoče se frekvence spektra prekrivati (aliasing).

### 9.14.3 Pravilno vzorčenje

Ko izvedemo t.i. sparse sampling moramo odstraniti vsebino z visoko frekvenco. To storimo z Gaussovim filtrom. Na ta način ponavljajoči se spektri vzorčenega signala se ne prekrivajo in ne poudarjajo napačnih frekvenc. Tako preprečimo aliasing.

## 9.15 Gaussova piramida

$$G_0 = Image$$

$$G_1 = (G_0 * gaussian) \downarrow 2 \text{ (smooth then subsample)}$$

$$G_2 = (G_1 * gaussian) \downarrow 2 \text{ (smooth then subsample)}$$

$$G_3 = (G_2 * gaussian) \downarrow 2 \text{ (smooth then subsample)}$$

$$G_4 = (G_3 * gaussian) \downarrow 2 \text{ (smooth then subsample)}$$

...

Nov nivo dobimo od prejšnjega tako, da gladimo z Gaussovim filtrom in naro vzorčimo. Gaussov filter uporabimo zaradi naslednje lastnosti:

$$G(\sigma_1^2) * G(\sigma_2^2) = G(\sigma_1^2 + \sigma_2^2)$$

Zelo dobro deluje z konvolucijo. Poleg tega je t.i. **low-band-pass filter**, zato dobimo nepotrebno reprezentcijo zgladene slike. Ni potrebe, da shranimo zgladeno sliko v polni resoluciji (razlog za spremembo velikosti).

### 9.15.1 Zakaj piramida?

- omogoča učinkovito implementacijo,
- omogoča večstopenjsko detekcijo objektov,
- omogoča večstopenjsko detekcijo robov, ...

## 10 Odvodi slik in detekcija robov