Spracovanie farebného obrazu

Kvantovanie

Mgr. Dana Škorvánková

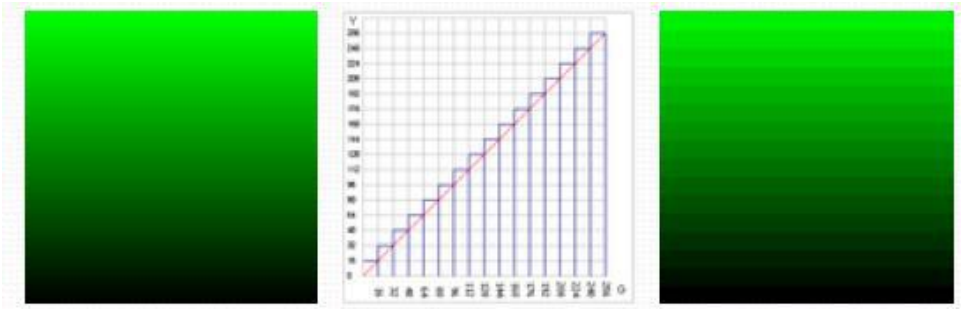
2022

Kvantovanie

- Čo je kvantovanie?

Kvantovanie

- Čo je kvantovanie?
  - diskretizácia hodnôt obrazovej funkcie (OF)
  - OF sa rozdelí na intervaly
  - obrázok s väčším počtom farieb → obrázok s menším počtom farieb

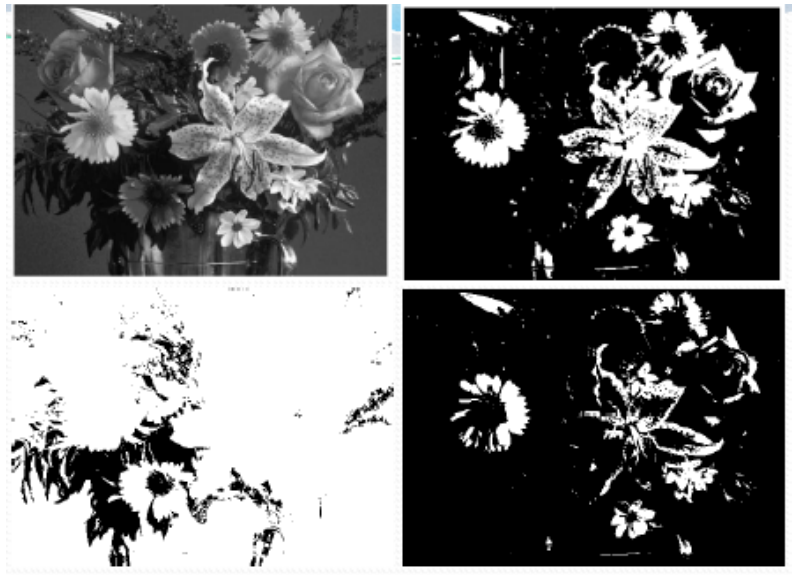


Kvantovanie

- Aký je najjednoduchší typ kvantovania?

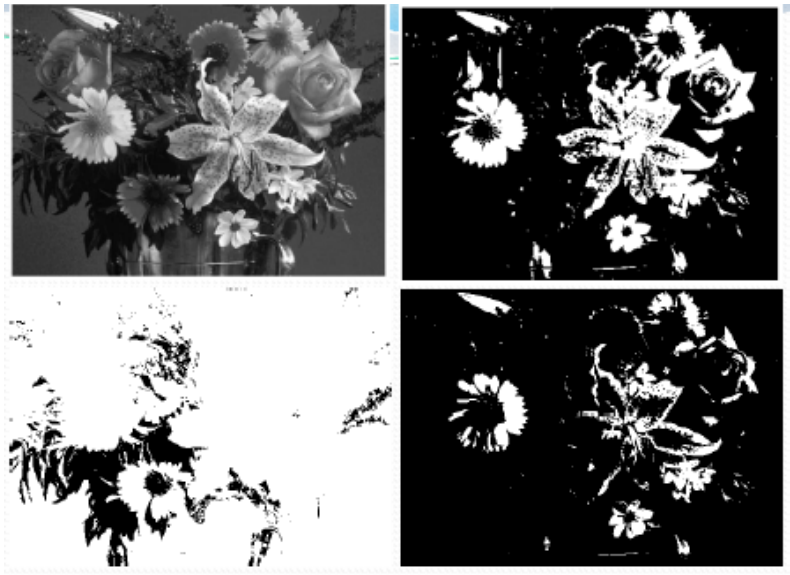
Kvantovanie

- Aký je najjednoduchší typ kvantovania?
  - Binarizácia
  - biela , čierna
- problém ?



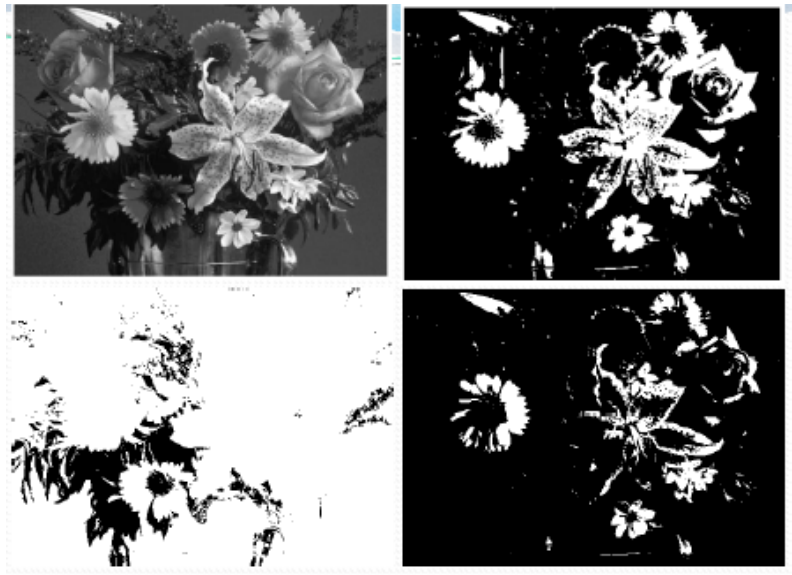
Kvantovanie

- Aký je najjednoduchší typ kvantovania?
  - Binarizácia
  - biela , čierna
- problém ?
  - strata informácie
- vylepšenie



Kvantovanie

- Aký je najjednoduchší typ kvantovania?
  - Binarizácia
  - biela , čierna
- problém ?
  - strata informácie
- Vylepšenie
  - určiť prah tak, aby sa zachovala priemerná intenzita



Kvantovanie

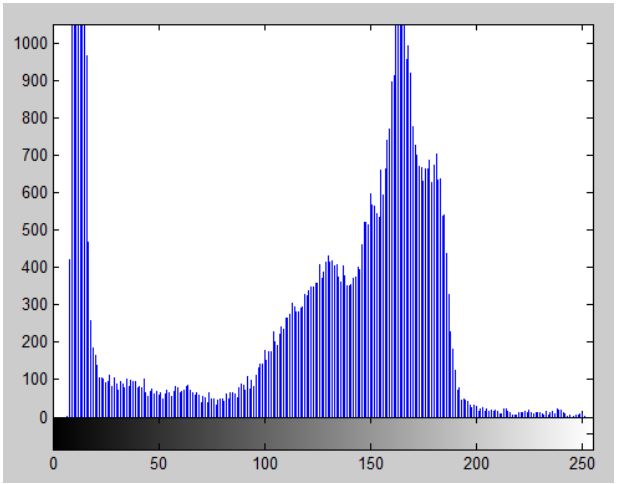
- určiť prah tak, aby sa zachovala priemerná intenzita
  - Napr. ak je priemerná intenzita vstupného obrazu 0,7 ➔ chceme, aby cca 70% pixelov malo intenzitu vyššiu, a 30% nižšiu.
  - ➔ 70% pixelov bude mať hodnotu 1 (po binarizácii) a 30% hodnotu 0.
  - ➔ priemer = (0,7**p***1 + 0,3**p***0) / **p** = 0,7**p**/**p** = 0,7
  - kde **p** je počet pixelov obrazu.
  - Zachovali sme pôvodnú priemernú intenzitu.

Prahovanie histogramu

- Ako vytvoríme histogram (šedotónového) obrazu?

Prahovanie histogramu

- Ako vytvoríme histogram (šedotónového) obrazu?
  - početnosti zastúpenia intenzitných úrovní v obraze



Prahovanie histogramu

- Ako vytvoríme histogram (šedotónového) obrazu?
  - početnosti zastúpenia intenzitných úrovní v obraze

- V Matlabe?

Prahovanie histogramu

- Ako vytvoríme histogram (šedotónového) obrazu?
  - početnosti zastúpenia intenzitných úrovní v obraze

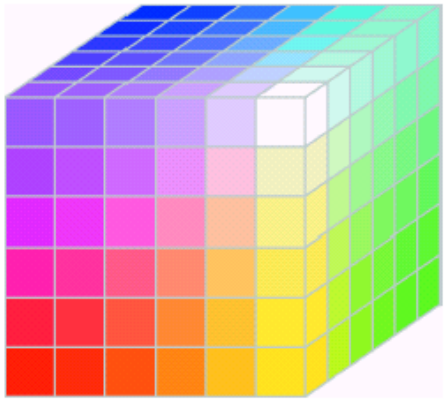
- V Matlabe?
  - `imhist(img)`
    - vykreslí histogram
  - `h = imhist(img);`
    - uloží histogram do premennej h
  - `[h, levels] = imhist(img);`
    - vráti histogram aj príslušné intenzitné úrovne

Kvantovanie – binarizácia

- Načítajte obrázok `peppers.png`
  1. Prekonvertujte ho na šedoúrovňový a pomocou `imhist` zistite počet pixlov pre jednotlivé úrovne šedej (pracujte s obrazom s hodonotami v intervale 0-1)
  2. Zistite priemernú intenzitu `m` šedoúrovňového obrázka
  3. Vypočítajte pre každú intenzitnú úroveň histogramu, koľko percent pixelov obrazu má nižšiu alebo rovnakú intenzitu - pomocou funkcie `cumsum` (kumulatívna suma)
  4. Nájdite index príslušného prahu (`find`), kde budete brať do úvahy priemernú intenzitu pôvodného obrázka (prah bude nad `1-m` percentami pixelov s najnižšou intenzitou)
  5. Vyprahujte výsledný obrázok a zistite jeho priemernú intenzitu.

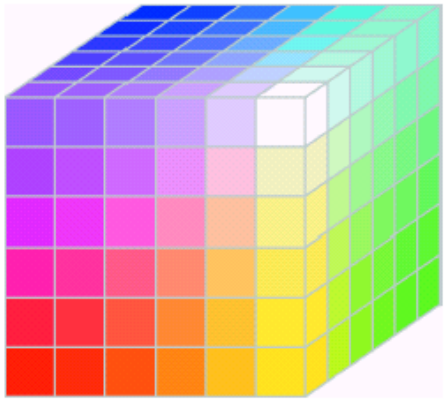
Uniformné kvantovanie

- Rozdelí RGB priestor (kocku) rovnomerne na menšie kocky rovnakej veľkosti



Uniformné kvantovanie

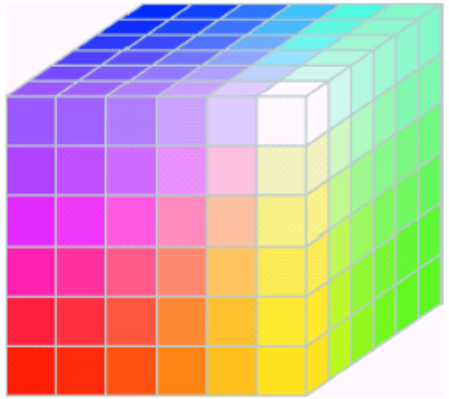
- Rozdelí RGB priestor (kocku) rovnomerne na menšie kocky rovnakej veľkosti
- Neprispôsobuje rozdelenie farieb ich zastúpeniu v pôvodnom obraze



Uniformné kvantovanie

- Rozdelí RGB priestor (kocku) rovnomerne na menšie kocky rovnakej veľkosti
- Neprispôsobuje rozdelenie farieb ich zastúpeniu v pôvodnom obraze

- `[X, map] = rgb2ind(img, tolerance);`
- `tolerance` = veľkosť malej kocky (normalizovaná na 0-1)
- `X` obsahuje pre každý pixel label farby (id)
- `map` obsahuje RGB farbu pre každý label (id)
- `imshow(X, map)`

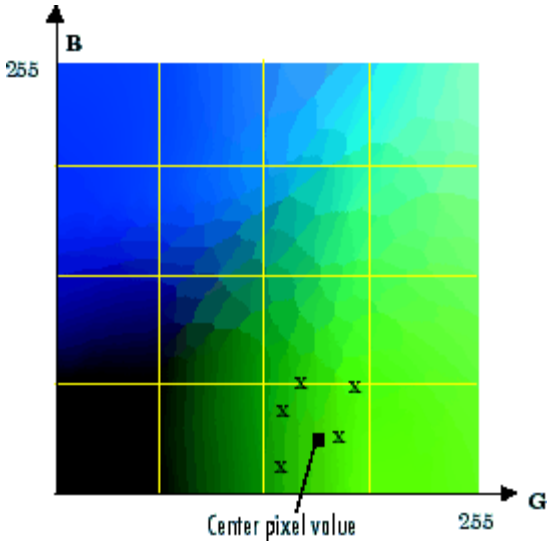


Minimum variance quantization

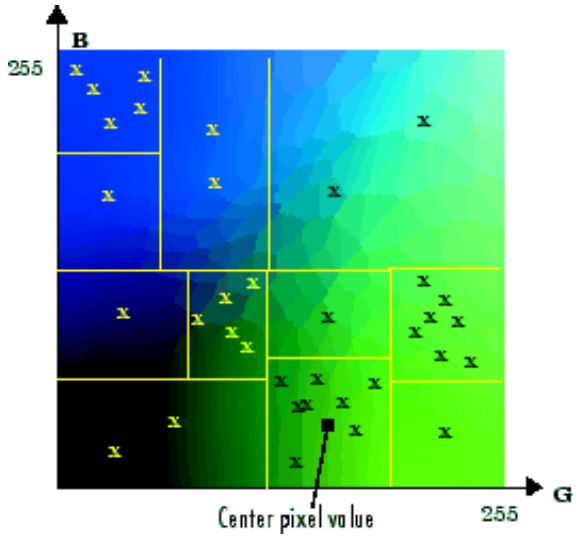
- Zohľadňuje zastúpenie farieb v obraze
  - Lepší výsledok než uniformné kvantovanie
  - Pre farby s väčším zastúpením vytvorí väčší chlievik (box)

Minimum variance quantization

- Zohľadňuje zastúpenie farieb v obraze
  - Lepší výsledok než uniformné kvantovanie
  - Pre farby s väčším zastúpením vytvorí väčší chlievik (box)



Uniformné kvantovanie



Kvantovanie s minimálnou varianciou

Minimum variance quantization

- Zohľadňuje zastúpenie farieb v obraze
  - Lepší výsledok než uniformné kvantovanie
  - Pre farby s väčším zastúpením vytvorí väčší chlievik (box)
  - Zgrupuje pixely s nízkou varianciou farieb

Minimum variance quantization

- Zohľadňuje zastúpenie farieb v obraze
  - Lepší výsledok než uniformné kvantovanie
  - Pre farby s väčším zastúpením vytvorí väčší chlievik (box)
  - Zgrupuje pixely s nízkou varianciou farieb

  - `[X, map] = rgb2ind(img, max_n_colors);`
  - `max_n_colors` = maximálny počet farieb (chlievikov) vo výstupnej colormape

Minimum variance quantization

- `[X, map] = rgb2ind(img, max_n_colors);`
- `max_n_colors =` maximálny počet farieb (chlievikov) vo výstupnej colormape

- Funkcia `rgb2ind` defaultne aplikuje dithering
  - Na zvýšenie počtu zdanlivo odlišných farieb vo výstupe
  - Mení farby pixlov v rámci lokálneho okolia tak, aby sa cca zachovala ich priemerná farba z pôvodného obrazu

- `[X, map] = rgb2ind(img, max_n_colors, 'nodither');`
  - Výslednok bez ditheringu

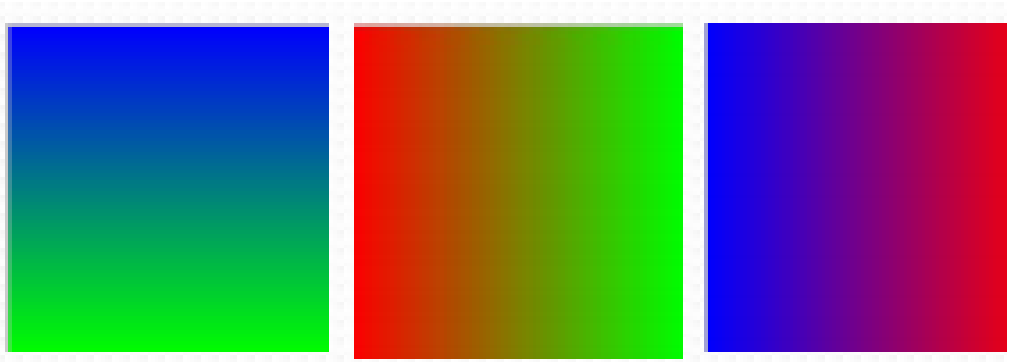
K-means kvantovanie

- script `Kmeans.m`
- `pdist2.m`
- k = 20 (farieb)

Úloha 1 – vytvorte funkcie

Nepoužívajte hotové funkcie na gradient, alebo konverziu, ani ich negooglite!

1. gradient(color1, color2, direction) (2b)*
   - modrá – zelená, červená – zelená, modrá – červená …



2. conversion(img) (1b)*
   - preveďte obraz z RGB do CMY (netreba vykreslovať)

*pri odovzdaní na cviku, inak spolu max. 1b