Ppgso prednaska c2

**Čo je obraz?**

Obdlznikove 2D pole pixelov

-pixely NIE su male stvorceky – pixely su vzorky z realneho priestora ktora sa snazi zachytit realny svet/priestor

- pre programatora je to struktura pamate

- obvykle reprezentovana ako postupnost pixelov

- typicky riadok po riadku zlava do prava

**Spracovanie obrazu**

* Filtrovanie obrazu / Manipulacia s obrazom
  + Operacia na urovni pixelov
  + Filtrovanie
  + Kompozicia
  + Kvantizacia
  + Deformacia a morfovanie
  + Vzorkovanie, rekonstrukcia a alias

**Operacie na urovni pixelov**

* Zmen hodnoty farby pixelu funkciou F
  + P´ = F (P)
* Napriklad pridat nahodnu hodnotu do kazdeho kanalu farby (pridanie sumu)
* Orezat (Clamp) na rozsah <0,1>
* Preskalovat hodnoty pixelu a orezat na <0,1> (zmena jasu)
* Zmena kontrastu – pouziva sa vypocet stredneho jasu L = 0.3r+0.59g+0.11b
  + Preskalovat odchylku od L a orezat na <0,1>
* Odtiene sedej (greyscale) – r´, g´, b´ = 0.3+0.59g+0.11b

**Lineárne filtrovanie**

* Kazdy pixel je linearnou kombinaciou pixelov v jeho okoli
* Diskretizovana konvolucia funkcii
  + Kazdy pixel je linearnou kombinaciou pixelov vo svojom okoli
  + Ako vyriesit okrajove pixely?
* Rozmazanie (blur)
  + Konvolucia pomocou filtra, ktoreho suma je 1
  + Je to nejaka matica, pomocou ktorej dostaneme rozmazanie
* Detekcia hran
  + Konvolucia s filtrom najde rozdiel medzi pixelmi
* Zaostrenie (Sharpen)
  + Spocitanie hran s popvodnym obrazom
* Stale je to ten isty algoritmus, lens a menia matice

**Nelinearne filtrovanie**

* Kazdy pixel je nelinearnou funkciou vstupnych pixelov
* Nelinearny filter – neda sa urobit konvoluciou
* Median NIE JE linearny

**Miešanie (blending)**

* Kombinacia viacerych obrazov a vytvorenie noveho obrazu
* Per-pixel operacia
* Blend = F(base, source)
* Rozmery obrazu nemusia byt totozne

**Normalne miešanie (normal blending)**

* Linearna kombinacia pixelov z oboch obrazkov
* Blend(x,y) = base(x,y)(1-t) + source(x,y)t
* T je v rozsahu <0,1>

**Aditivne miešanie**

* Scita pixely z oboch obrazkov
* Oreze vysledok na rozsah <0,1>

**Odcitanie**

-odcitanie pixely z oboch obrazkov

**Rozdiel**

* Vypocita rozdiel medzi obrazkami
* Biela farba v source invertuje farbu v base

**Maskovanie**

* Ako spojit casti obrazu?
* Potrebujeme sposob ako vybrat casti obrazku
* Napad: co tak pouzit iny obrazok
* **Maska**
  + Biela oblast je priehladna
  + Cierne oblasti su nepriehladne
  + Na vyber oblasti zaujmu sa pouzije 1-bitovy obraz
* Problemy masky
  + Masky su 1-bitove, bez hladkych okrajov
  + Ako riesit priehladne predmety, ako je sklo?
  + Potrebujeme pracovat s dodatocnym obrazkom?

**Alpha blending**

* Napad: ukladat priehladnost pixelov per pixel
* Pixel = (r, g, b, a), pridany alpha kanal
* A definuje priehladnost pixelu, 1 priehladny, 0 nepriehladny

**Priklady masky, priesvitnost**

* Photoshop masky a vrstvy
* Viacnasobne rezimy prelinania a priehladnost
* Greenscreen

**Kvantizacia**

* Artefakt v dosledku obmedzeneho rozlisenia intensity
  + Vyrovnavacie pamate snimok (framebuffer) maju obmedzeny pocet bitov na pixel

**Redukcia efektov kvantizacie**

* Poltonovanie
* Dithering (rozptylovanie
  + Nahodne
  + Usporiadane
  + Chybove difuzne

Klasické poltonovanie

* Vyjadrenie intensity sa pouzije velkost bodu
* Plocha bodov je umerna intenzite v obrazku

Poltónové vzory

* Na vyjadrenie intensity sa pouzije zhluk pixelov
* Vymiena priestorove rozlisenie za rozlisenie intensity
* Kolko intenzit je v n x n?

Dithering

* Distribuuje chyby medzi pixely
  + Vyuziva priestorovu integraciu v ludskom oku
  + Zobrazuje vacsi rozsah vnimatelnych intenzit
* Nahodne rozptylovanie
  + Randomizuje chyby kvantizacie
    - Chyby sa prejavia ako šum
* Usporiadane rozptylovanie
  + Pseudonahodne chyby kvantizacie
  + Matica reprezentuje vzory prahov
  + Vysledky mierne lepsie ako pri nahodnom
* Chybove difuzne rozptylovanie
  + Chyby su distribuovane na pixely nachadzajuce sa vpravo a nizsie
  + Robert W. Floyd a Louis Steinberg, 1976
  + Vysledok je pomerne dost dobry

**Warping**

* Transforuje obrazove pixely
* Mapovanie

**Mapovanie**

* Definuje transformaciu obrazu
  + Definuje ciely pre kazde miesto v zdroji (alebo naopak, ak je to invertovatelne!

**Skalovanie**

* Skalovanie zvacsit zmensit

**Rotacia**

* Rotacia o uhol

**Skosenie**

* Skosenie v x smere o factor, y sa zachovava alebo co idk

**Implementacia**

* Dopredne mapovanie
  + Iteruje cez zdrojovy obrazok
  + Mnoho pixelov sa mapuje na rovnaky ciely!
  + Niektore pixely nebudu pokryte!
  + Moze proste produkovat artefakty
* Inverzne mapovanie
  + Vypocitava nejakov inverznou funkciou tie koordinaty
  + Iteruje cez cielovy obrazok
    - Musi prevzorkovat zdrojovy obrazok
    - Ovela jednoduchsie, ale moze sa nad-vzorkovat (oversampling
  + Prevzorkovanie
    - Vyhodnocuje zdrojovy obrazok na lubovolnom mieste
    - Suradnice vo vseobecnosti nie su cele cisla
  + Najblizsi sused
    - Hodnota najblizsieho pixelu
    - Rychle ale nizka kvalita
* Bilinearne filtrovanie
  + Bilinearne interpolujte styri najblizsie pixely
    - a = linearna interpolacia medzi src(u1,v2) a src(u2,b2
    - To iste pre b
    - Primerane rychle aj kvalita je fajn
* Ine filtre
  + Bikubicke filtrovanie
    - Berie do uvahy 4x4 pixely
    - Hladsie, menej artefaktov
    - Vypoctovo nakladne
  + Gaussovo filtrovanie
    - Pouziva vazeny sucet okolia
    - Vahy sa normalizuju pomocou gaussovej funkcie