

Skeletonizace a ztenčování

KIV/ZVI

Ondřej Vaic

A17B0385P

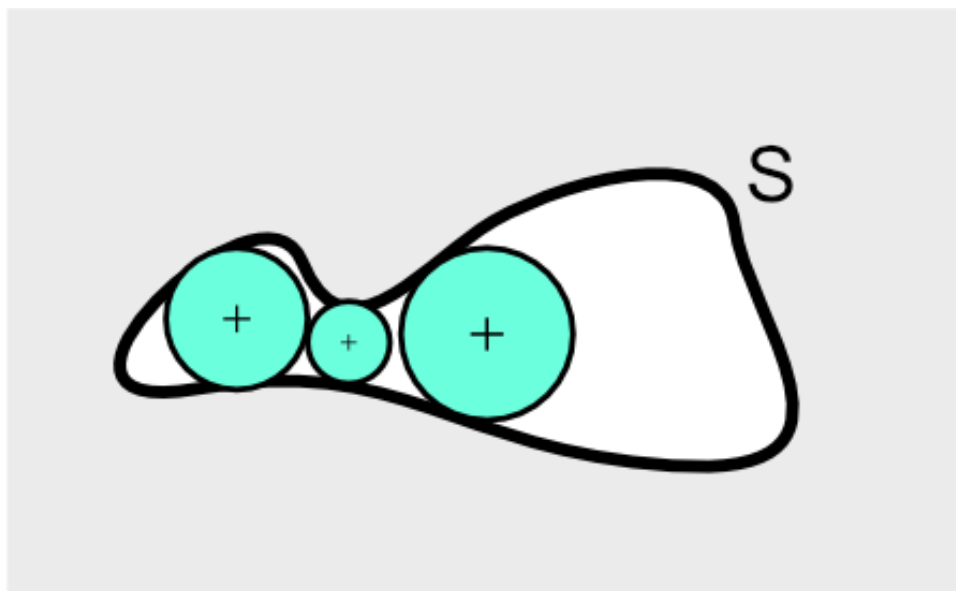
ondra.vaic@gmail.com

Fakulta aplikovaných věd
Západočeská Univerzita
06/05/2020

1 Úvod

Skeletizace a ztenčování

Skeletizace je proces vytvoření čárového obrazu z binárního obrazu. Cílem je vytvořit obraz, který zachovává původní tvar, ale je tvořen pouze tenkými čarami. Takový obraz může být jednoduší dále analyzovat. Skelet oblasti S je definován jako množina všech středů kružnic vepsaných S , které se dotýkají dvou a více bodů hranice S . Pro kružnice navíc platí, že neexistuje žádná kružnice s větším poloměrem a stejným středem, která by do S patřila, viz Obrázek 1.



Obrázek 1: Ilustrace definice skeletu oblasti.

2 Algoritmus

V této práci byla implementována metoda hledání skeletu metodou mediální osové transformace (anglicky Medial Axis Transformation). Ta vychází z představy oblasti jako trávníku. Trávník je zapálen po celé obvodu v jeden okamžik, oheň se poté šíří konstantní rychlostí. Body kde se potkají dva a více ohňů jsou považovány za skelet.

3 Implementace

Implementace je založena na vytvoření vzdálenostní transformace. Dále bude vysvětlen výpočet vzdálenostní transformace. V tomto algoritmu je dvakrát

použito okolí bodu, různé kombinace okolí budou dávat různé výsledky viz rozbor dosažených výsledků.

3.1 Vzdálenostní transformace

Předpokládejme že obrazový bod popředí je 1, obrazový bod pozadí je 0. Pro všechny body obrazu, je hodnota bodu v následující iteraci rovna součtu hodnoty původního bodu a minima z hodnot bodů v jeho okolí. Algoritmus končí když je obraz oproti předchozí iteraci stejný. Následuje ukázka implementace v programovacím jazyku python. Zde `newImg` bude výsledný obraz vzdálenostní transformace, `originalImage` je binární obraz ke kterému hledáme skelet a `currentImage` je inicializován jako kopie `originalImage`.

```
while True:
    newImg = np.copy(currentImage)

    for y in range(1, image.shape[0] - 1):
        for x in range(1, image.shape[1] - 1):
            newImg[y][x] = originalImage[y][x] +
                min(neighbourhood(currentImage, y, x))

    if isIdentical(currentImage, newImg):
        break;

    currentImage = newImg
```

Na následujícím obrázku lze vidět vizualizaci obrazu vzdálenostní transformace jako šedotónový obraz, zde byl vstupní obrázek obdélník.



Obrázek 2: Vizualizace obrazu vzdálenostní transformace obdélníku.

3.2 Optimalizace

Pro optimalizaci tohoto algoritmu je důležité uvědomit si, že body které se oproti předchozí iteraci nezmění, se nezmění ani po zbytek výpočtu algoritmu. Souřadnice nezměněných bodů lze ukládat do datové struktury množina, která využívá hashování, to zaručí rychlé ověření zda prvek do množiny patří. Body, které do této množiny neměnných prvků patří už není nutné přepočítávat. Následuje optimalizovaná verze algoritmu

```

unchanged = set()

while True:
    newImg = np.copy(currentImage)
    changed = False

    for y in range(1, image.shape[0] - 1):
        for x in range(1, image.shape[1] - 1):
            if (y, x) in unchanged:
                continue

            lastVal = currentImage[y][x]
            newImg[y][x] = originalImage[y][x] +
            min(neighbourhood(currentImage, y, x))

```

```

        if lastVal == newImg[y][x]:
            unchanged.add((y, x))
        else:
            changed = True

currentImage = newImg

if not changed:
    return currentImage

```

3.3 Nalezení skeletu

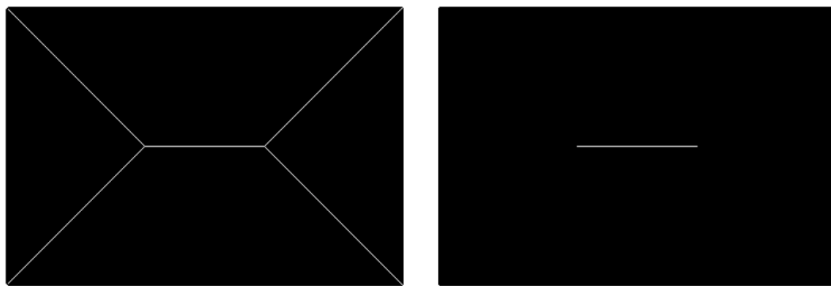
Z obrazu vzdálenostní transformace jsou body skeletu určeny jako body, které mají vůči svému okolí maximální hodnotu. Následuje ukázka kódu v programovacím kódu python, která vytvoří z obrazu vzdálenostní transformace `img` obraz skeletu jako `newImg`.

```

newImg = np.zeros(img.shape, np.uint8)
for y in range(1, img.shape[0] - 1):
    for x in range(1, img.shape[1] - 1):
        maximum = max(neighbourhood(img, y, x))
        if img[y][x] >= maximum:
            newImg[y][x] = 255
        else:
            newImg[y][x] = 0

```

Na následujícím obrázku lze vidět skelety, které vzniknou z obrazu vzdálenostní transformace, který lze vidět na obrázku 5. Nalevo je určen skelet pomocí osmi okolí, napravo pomocí čtyř okolí.

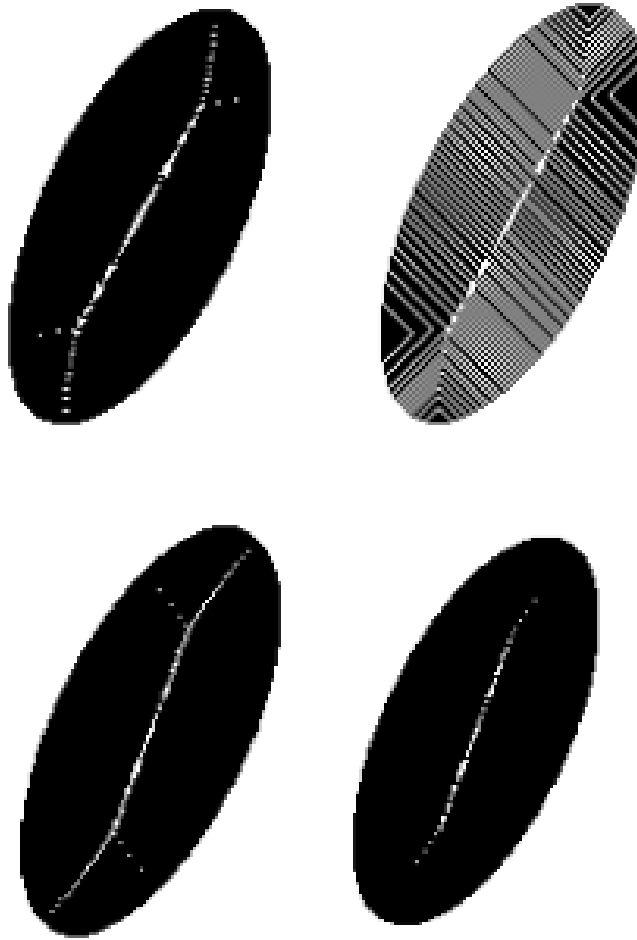


Obrázek 3: Skelety obdélníku, nalevo skelet určen čtyř okolím, napravo osmi okolím.

4 Výsledky algoritmu

4.1 Okolí

Při implementaci algoritmu záleží, jaké okolí je zvoleno pro vzdálenostní transformaci δ_1 a pro určení skeletu δ_2 . Na následující obrázcích lze vidět různé volby kombinace zvolených okolí a výsledky skeletizace obrázku nakloněné elipsy.

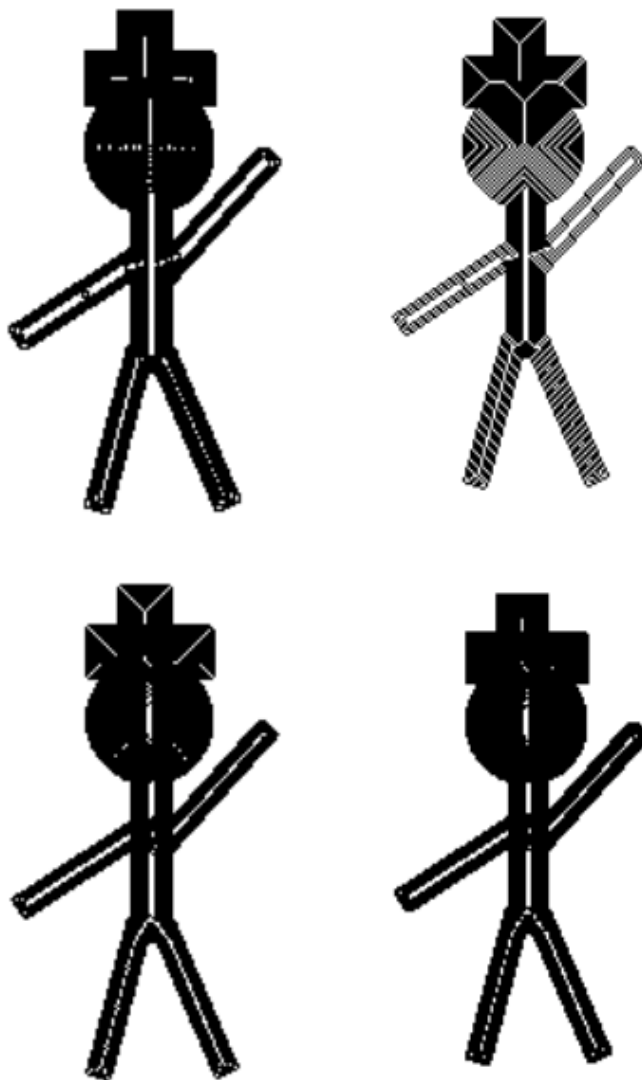


Obrázek 4: Nahoře vlevo $\delta_1 = 8 - \delta$, $\delta_2 = 8 - \delta$; nahoře vpravo $\delta_1 = 8 - \delta$, $\delta_2 = 4 - \delta$; dole vlevo $\delta_1 = 4 - \delta$, $\delta_2 = 4 - \delta$; dole vpravo $\delta_1 = 4 - \delta$, $\delta_2 = 8 - \delta$

Je patrné, že kombinace $\delta_1 = 8 - \delta$, $\delta_2 = 4 - \delta$ není vhodná. Obdobné výsledky jsou i pro jiné obrázky. Nejlepší skelet je určen kombinací $\delta_1 = 4 - \delta$,

$\delta_2 = 4 - \delta$. Další dvě kombinace poskytují podobný výsledek, který obsahuje dva falešné skelety.

Na následujícím obrázku lze vidět znovu výsledky pro různé kombinace okolí.



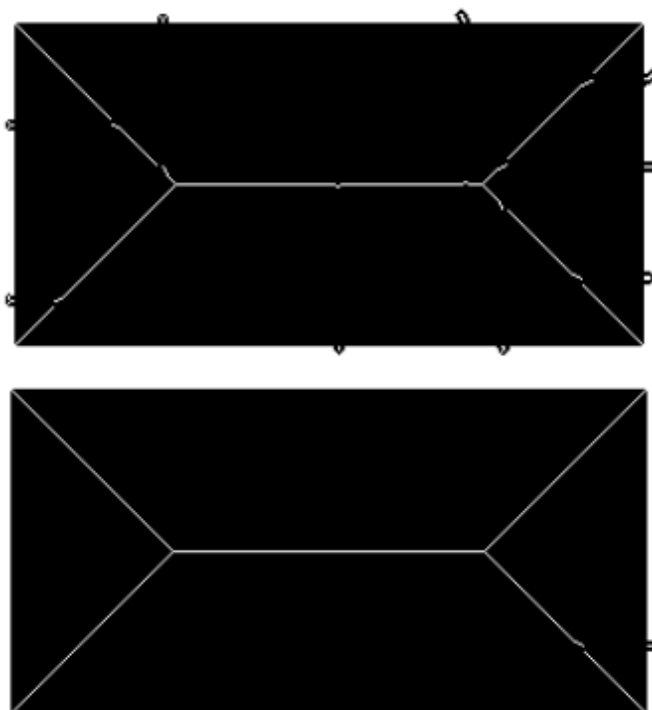
Obrázek 5: Nahoře vlevo $\delta_1 = 8 - \delta$, $\delta_2 = 8 - \delta$; nahoře vpravo $\delta_1 = 8 - \delta$, $\delta_2 = 4 - \delta$; dole vlevo $\delta_1 = 4 - \delta$, $\delta_2 = 4 - \delta$; dole vpravo $\delta_1 = 4 - \delta$, $\delta_2 = 8 - \delta$

Kombinace $\delta_1 = 8 - \delta$, $\delta_2 = 4 - \delta$ má znovu špatné výsledky. Nejlepší je v tomto případě nejspíše $\delta_1 = 8 - \delta$, $\delta_2 = 8 - \delta$. Tato kombinace v tomto případě zachovává původní tvar, a většina skeletu je navíc spolu propojena.

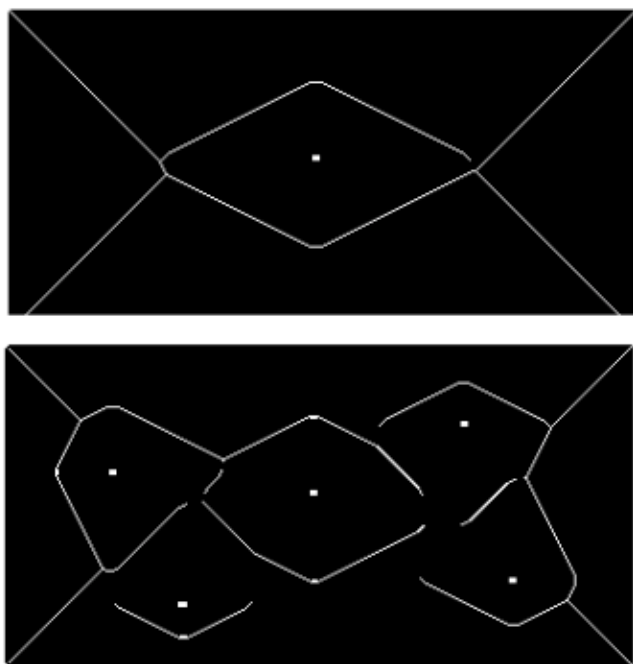
Nejllepší volba okolí záleží na vstupním obrázku, a na tom k čemu výsledek potřebujeme. Jednoznačně se dá ale říci, že $\delta_1 = 8 - \delta$, $\delta_2 = 4 - \delta$ není dobrá volba.

4.2 Díry a výběžky

Implementovaný algoritmus je citlivý na díry ve zpracovaném snímku a výběžky po jeho obvodu. To jsou artefakty, které mohou vzniknout segmentací. Dále budou ukázky některých obrázků a jak je jejich skeleton ovlivněn těmito artefakty.

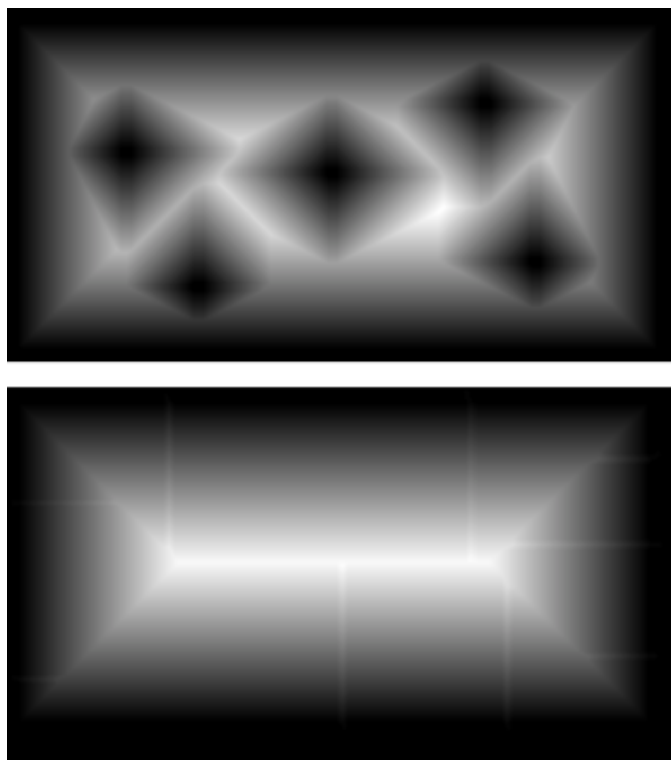


Obrázek 6: Skeletonizace obdélníka s jedním, a více výběžky po obvodu.



Obrázek 7: Skeletonizace obdélníka s jednou, a více děrami.

Lze vidět že díry v obrazu mění výsledný skeleton více, než výběžky. Na následujícím obrázku lze vidět vizualizace vzdálenostní transformace. Z ní je patrné proč díry ovlivňují výsledný skeleton dramatictější než výběžky.



Obrázek 8: Vizualizace vzdálenostní transformace obdélníka s dírami a obdélníka s výběžky.

Artefakty, které byly výše uvedené je dobré eliminovat v předzpracování snímku. Dále budou uvedeny některé metody, které to umožňují.

5 Předzpracování

Skeletizaci je možné provádět pouze s binarizovaným obrazem. Před samotným zpracováním je tedy nutné obraz segmentovat na popředí a pozadí. Segmentovaný obraz může ale obsahovat díry a nebo výběžky po obvodu objektu. Algoritmus hledání skeletu je na tyto chyby citlivý, zejména na díry, je proto dobré se těchto artefaktů zbavit.

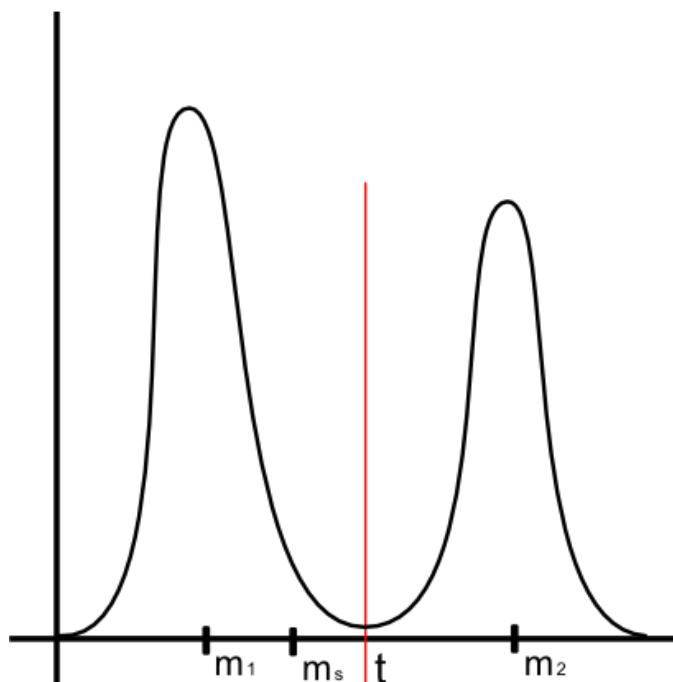
5.1 Segmentace

V této práci byla segmentaci docílena metodou OTSU. Je to metoda automatického hledání prahu. Vychází z předpokladu, že segmentovaný obraz má bimodální histogram. Práh je značen t a rozděluje obraz na dvě třídy. Určíme m_s jako střední hodnotu histogramu snímku. Dále m_1 a m_2 jako střední hodnoty

dvou histogramů, které vzniknou rozdělením histogramu v bodě t na dvě třídy. A nakonec pravděpodobnosti P_1 a P_2 , které určují jaká je pravděpodobnost, že náhodný bod patří do první resp. druhé třídy.

Práh t je poté nalezen tak, aby byla maximalizován rozptyl mezi třídami, který je dán jako.

$$\sigma^2 = (m_1 - m_s)^2 - (m_2 - m_s)^2$$



Obrázek 9: Ilustrace bimodálního histogramu, který je rozdělen metodou OTSU.

Na následujícím obrázku lze vidět nalevo šedotónový obrázek a napravo výsledek po segmentaci metodou OTSU. Výsledek obsahuje velké množství děr a výběžků. Další sekce se bude zabývat jejich odstraněním.



Obrázek 10: Šedotónový obraz a jeho segmentace metodou OTSU.

5.2 Eliminace děr a výběžků

Výše byly uvedeny problémy, které způsobí špatně segmentovaný obraz. Morfologické operace otevření a uzavření lze použít jako operace, které eliminují tyto artefakty, ale zachovávají celkový tvar.

Další možností je aplikování filtru Gaussovského rozostření před segmentací.

5.2.1 Morfologické otevření a uzavření

Operace morfologického otevření je aplikace eroze a následné aplikace dilatace. Uzavření je aplikace dilatace a poté eroze.

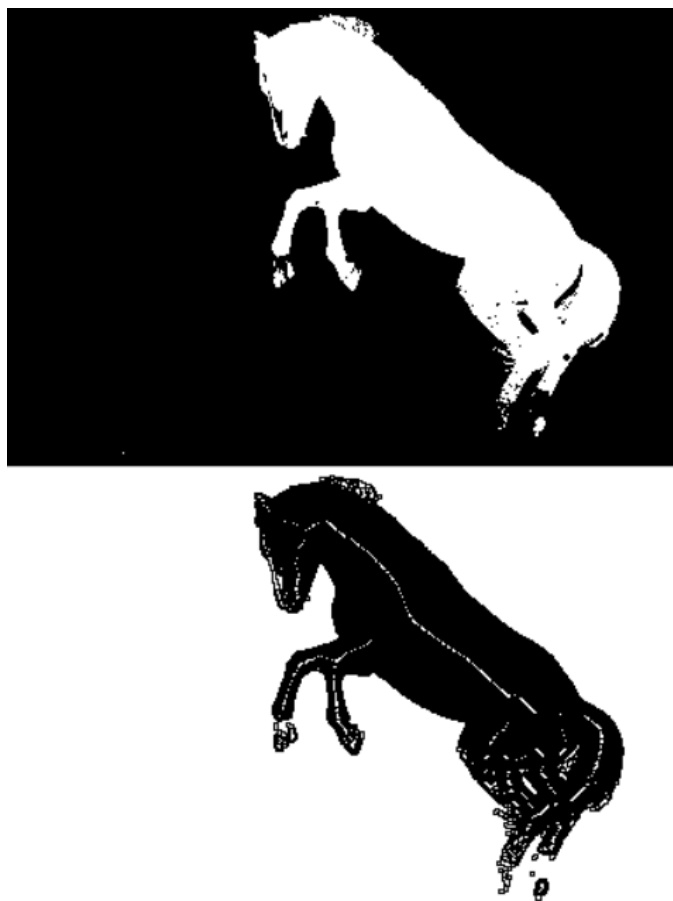
$$open(A, B) = (A \ominus B) \oplus B$$

$$close(A, B) = (A \oplus B) \ominus B$$

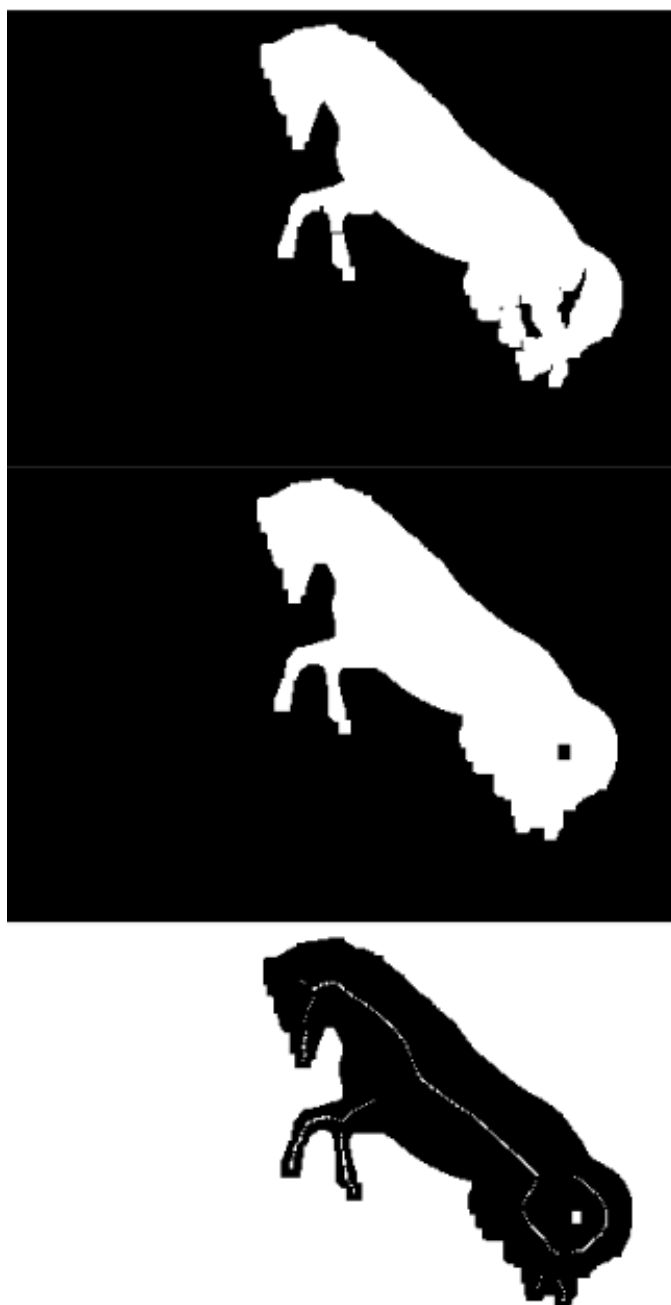
Otevření se hodí pro obrázek u kterého můžeme očekávat výběžky. Eroze nejdříve zmenší plochu obrazu a vymaže detaily, dilatace poté zvětší obraz zpět přibližně do stejného obsahu.

Uzavření se hodí pro obrázek ve kterém můžeme očekávat díry. Dilatace zaplní menší díry, ale zároveň zvětší plochu obrazu, eroze poté plochu zmenší zpět přibližně do stejného obsahu.

Na následujících obrázcích lze vidět výsledky skeletonu po různém předzpracování.



Obrázek 11: Nahoře obraz, který byl pouze segmentován metodou OTSU. Dole výsledný obraz po skeletonizaci.



Obrázek 12: Nahoře obraz, který byl po segmentaci otevřen. Uprostřed obraz který byl poté uzavřen. Jako strukturní element byl použit čtverec 7x7. Dole výsledný obraz po skeletonizaci.

5.2.2 Gaussovské rozmazání

Gaussovské rozmazání je konvoluční filtr, který rozmaže vstupní obraz. Lze zvolit jak velké okolí má být bráno v potaz pro každý pixel. Rozmazáním se zbavíme detailu a následná segmentace dá hladší plochu, která je vhodnější pro skeletizaci.



Obrázek 13: Nahoře obraz, který byl rozmazán. Filtr gaussovského rozmazání o šířce 9 byl aplikován 5x. Uprostřed segmentovaný obraz. Dole výsledný obraz po skeletizaci.

5.3 Shrnutí

Z obrázků výše je zřejmé, že je důležité obraz dobře předzpracovat před jeho skeletizací. Nejdříve je nutné obraz transformovat do šedotónového obrazu a následně segmentovat. Obraz je dobré buď rozmazat před segmentací, nebo ho po segmentaci otevřít a uzavřít.

6 Dosažené výsledky

V této práci byl vytvořen jednoduchý software, který umožňuje některé základní operace s obrazem. Software umožňuje nahrát obrázek a několik transformací pro předzpracování: transformovat do šedotónového obrazu, invertovat barvy, segmentovat prahem pomocí metody OTSU nebo ručním zvolením prahu, rozmazat gaussovským rozmazáním, základní morfologické operace, tedy eroze, dilatace, otevření a uzavření. Hlavní je, že software umožňuje skeletizaci algoritmem založeným na vzdálenostní transformaci.

Algoritmus byl otestován s různými kombinacemi pro volby okolí. Bylo zjištěno, že kombinace $\delta_1 = 8 - \delta$, $\delta_2 = 4 - \delta$ není použitelná. To jak vhodné jsou ostatní kombinace okolí záleží na vstupním obrazu.

Bylo zjištěno, že je nutná dobrá segmentace a následná úprava obrazu tak, aby obsahoval co nejméně děr, které by v obrazu neměly být a malých výběžků. Vhodné je obraz před segmentací rozmazat, nebo po segmentaci aplikovat morfologické otevření a uzavření.