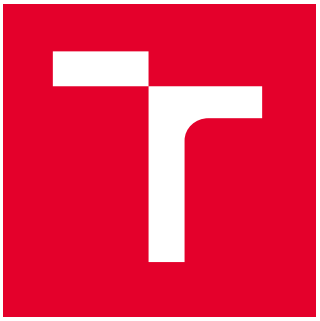


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MATEMATIKY

INSTITUTE OF MATHEMATICS

MĚŘENÍ GEOMETRICKÝCH PARAMETRŮ LIDSKÉ RUKY UŽITÍM NUMERICKÝCH METOD ZPRACOVÁNÍ OBRAZOVÉ INFORMACE

HUMAN HAND GEOMETRIC PARAMETERS MEASUREMENT USING NUMERICAL METHODS OF IMAGE PROCESSING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Anna Vanžurová**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Pavel Štarha, Ph.D.**
SUPERVISOR

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání diplomové práce

Ústav:	Ústav matematiky
Studentka:	**Bc. Anna Vanžurová**
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Matematické inženýrství
Vedoucí práce:	**doc. Ing. Pavel Štarha, Ph.D.**
Akademický rok:	2023/24

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Měření geometrických parametrů lidské ruky užitím numerických metod zpracování obrazové informace

Stručná charakteristika problematiky úkolu:

V potravinářském průmyslu se pravidelně sleduje hygiena výrobních procesů formou stěrů jak z povrchu výrobního zařízení, tak i stěry z rukou zaměstnanců. U stěrů ze zařízení je vždy známa celková plocha, ze které se stěr prováděl. V případě stěru z ruky zaměstnance, kdy se odebírá vzorek z celého jejího povrchu, celkovou plochu neznáme. Ve výsledcích testu se pak zjednodušeně uvádí počet bakterií na ruku, místo na jednotku plochy. Cílem této práce je najít numerickou metodu odhadu povrchu lidské ruky na základě zpracování obrazové informace. Tímto by bylo možné při hygienických testech uvádět množství bakterií na jednotku plochy stejně jako v ostatních testech výrobních zařízení. Jedná se o téma navazující na výsledky předcházející bakalářské práce.

Cíle diplomové práce:

Nastudovat a popsat základní numerické metody zpracování obrazové informace.
Určit vhodné geometrické parametry lidské ruky pro odhad jejího povrchu.
Provést měření na reálných datech a vyhodnotit získané výsledky.

Seznam doporučené literatury:

PRATT, William K. Digital Image Processing (Third Edition) PIKS Inside [online]. 3rd ed. New York: Wiley-Interscience, 2001 [cit. 2014-08-07]. ISBN 04-712-2132-5. Dostupné z: https://www.academia.edu/18481136/Digital_Image_Processing_Third_Edition_William_K_Pratt.

KLÍMA, Miloš. Zpracování obrazové informace. V Praze: České vysoké učení technické, 1996. ISBN 8001014363.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2023/24

V Brně, dne

L. S.

prof. RNDr. Josef Šlapal, CSc. ředitel ústavu	doc. Ing. Jaroslav Katolický, Ph.D. děkan fakulty

Abstrakt
Tato práce popisuje matematickou metodu odhadu povrchu lidské ruky. Užitím teorie zpracování obrazové informace, analytické geometrie a numerických metod byl vytvořen model pro zpracování fotografie ruky, pomocí které se vypočítá povrch ruky v softwaru Matlab. Model využívá nástroje jako je například mediánový filtr, adaptivní prahování nebo podmíněná eroze. Zajímavé využití tu má křivost křivky, pomocí které nalezneme vrcholy a úpatí prstů na hranici ruky. Mezi získanými výsledky modelu a změřenými reálnými povrchy rukou byla zjištěna závislost a byla nalezena regresní funkce, která udává bodový odhad povrchu lidské ruky.

Summary
This thesis deals with a matematic model used for estimation of the surface of a human hand. By using the theory for digital image processing, analytic geometry and numerical methods a model for proccesing a photografy of a hand was created. This model further calculates the surface in the software Matlab. The model uses tools such as median filter, adaptive tresholding or conditional erosion. Interesting is the usage of the curvature of a curve, which can be used to find the peaks and bases on the border of the hand. Dependencies between the calculated results and the measured real surfaces can be found. Therefore a regression function could be found, which gives us a point estimation of the human hand surface.

Klíčová slova
povrch ruky, zpracování obrazové informace, adaptivní prahování, regresní analýza, korelační analýza, křivost křivky

Keywords
human hand surface, numerical methods of image processing, adaptive tresholding, regression analysis, corelation analysis, curvature of a curve

VANŽUROVÁ, Anna. *Měření geometrických parametrů lidské ruky užitím numerických metod zpracování obrazové informace* [online]. Brno, 2024 [cit. 2024-05-04]. Dostupné z: https://www.vut.cz/studenti/zav-prace/detail/154134. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav matematiky. Vedoucí práce Pavel Štarha. Počet stran: 85 s.

Prohlašuji, že jsem svou diplomovou práci na téma *Měření geometrických parametrů lidské ruky užitím numerických metod zpracování obrazové informace* vypracovala samostatně pod vedením doc. Ing. Pavla Štarhy, Ph.D., s použitím uvedené literatury.

Bc. Anna Vanžurová

Děkuji svému vedoucímu práce doc. Ing. Pavlu Štarhovi, Ph.D. za trpělivost a plno cenných rad. Dále bych chtěla poděkovat Mgr. Janě Procházkové, Ph.D. za pomoc při skenování sádrových odlitků rukou.

Bc. Anna Vanžurová

Obsah

Úvod

Při zkoumání čistoty ve výrobních procesech potravinářského průmyslu se pořizují stěry bakterií z ploch výrobních zařízení, které se uvádějí v počtu bakterií na jednotku plochy. Kromě toho se pořizují i stěry z rukou zaměstnanců. U těchto stěrů není známa plocha, a tak mají odlišnou jednotku - počet bakterií na ruku. Proto se zde nabízí myšlenka, nalézt vhodnou metodu k získání odhadu povrchu lidské ruky. A právě tím se zabývá tato práce. Jedná se o práci navazující na bakalářskou práci s názvem *Měření geometrických parametrů lidské ruky* z které vychází i princip navrženého matematického modelu.

Navržená metoda odhadu povrchu lidské ruky, jejíž vstupem je fotografie ruky a výstupem je její povrch, využívá především numerických metod zpracování obrazové informace, které jsou podrobně popsány v kapitole 3. Například pomocí mediánového filtru a adaptivního prahování jsme schopni fotografii rozdělit na objekt a jeho pozadí. Morfologické operace pak využijeme například k definování uceleného objektu ruky v kapitole 4.1.2 díky čemuž dokážeme určit jednopixelovou hranici objektu.

Kromě nalezení objektu ruky se v práci dále zabýváme její geometrií. Abychom odhadli povrch ruky, je třeba její jednotlivé části aproximovat geometrickými útvary a tělesy. K tomu je například nutné nalézt tzv. *body zlomu* popsané v kapitole 4.2.1, díky nimž objekt rozdělíme na potřebné části a získáme o něm více informací. Tyto body nalezneme v místech největší lokální křivosti hranice objektu ruky. Proto se práce v úvodu věnuje tématům křivek a aproximaci funkcí (viz kapitoly 1.2, 1.3).

Důležitou součástí práce je statistické ověření metody a nalezení vhodného regresního modelu. Proto kapitola 2 obsahuje vybrané informace ze statistiky. Na konci praktické části práce se nachází statistické výsledky, porovnávající přesné povrchy deseti testovacích rukou s povrchy vypočtenými touto metodou. Přesným povrchem se zde myslí povrch získaný 3D skenem, který naskenuje sádrový odlitek ruky. Tímto způsobem získáme povrch, který se přibližně blíží k reálnému povrchu, avšak pro naše účely stačí a budeme ho považovat za reálný povrch.

1. Matematický aparát

Pro praktickou část práce je potřeba zavést několik důležitých definic z různých oblastí matematiky.

1.1. Gradient a Laplaceův operátor

Tato kapitola připomene dva důležité pojmy z oblasti diferenciálního počtu funkcí více proměnných, které jsou podrobněji popsány v [5] a [11].

Definice 1.1.1. Funkci $u \equiv u(x_1, x_2, \cdots, x_n)$, definovanou v určité oblasti $\Omega \subset \mathbb{R}^n$, nazýváme *skalárním polem*.

Poznámka 1.1.1. Plochy $u = konst$ jsou tzv. *hladiny* skalárního pole.

Gradient a Laplaceův operátor jsou diferenciálními operátory, které se využívají ve fyzikálních a geometrických úvahách. Veškeré následující definice lze zobecnit na n-dimenzionální případ, ale vzhledem k praktickému využití, jsou v tomto textu definovány pouze ve třírozměrném prostoru. Tedy nezávislé proměnné x_1, x_2, x_3 představují kartézské proměnné x, y, z.

Definice 1.1.2. Nechť $u \equiv u(x, y, z)$ je skalární pole a x, y, z jeho kartézské souřadnice. Pak vektor

$$\operatorname{grad} u = \frac{\partial u}{\partial x}\mathbf{i} + \frac{\partial u}{\partial y}\mathbf{j} + \frac{\partial u}{\partial z}\mathbf{k},$$

kde $\mathbf{i}, \mathbf{j}, \mathbf{k}$ jsou vektory ortonormální báze, se nazývá *gradient* skalárního pole u.

Gradient skalárního pole v bodě $[x_0, y_0, z_0]$ je tedy vektor, který je vždy kolmý k hladině procházející bodem $[x_0, y_0, z_0]$. Jedná se o jakousi nadstavbu parciální derivace, pomocí které lze sledovat trend pouze v omezeném počtu směrů. Zatímco gradient je vektor, který v každém bodě skalárního pole popisuje směr největšího růstu tohoto pole.

Poznámka 1.1.2. Pro zápis gradientu $\operatorname{grad} u$ definovaného v kartézských souřadnicích x, y, z se často využívá Hamiltonův operátor nabla, který je definovaný jako $\nabla = \mathbf{i}\frac{\partial}{\partial x} + \mathbf{j}\frac{\partial}{\partial y} + \mathbf{k}\frac{\partial}{\partial z}$. Pak lze psát

$$\operatorname{grad} u = \nabla u = \left(\mathbf{i}\frac{\partial}{\partial x} + \mathbf{j}\frac{\partial}{\partial y} + \mathbf{k}\frac{\partial}{\partial z}\right) u.$$

Definice 1.1.3. Skalárním násobením operátoru ∇ se sebou samým získáme tzv. *Laplaceův operátor* Δ (delta):

$$\Delta = \nabla \cdot \nabla = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}.$$

Poznámka 1.1.3. Aplikuje-li se Laplaceův operátor na skalární pole u, tj. $\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}$, výsledkem je znovu skalární pole.

1.2. Křivky

Následující informace jsou čerpány z [4].

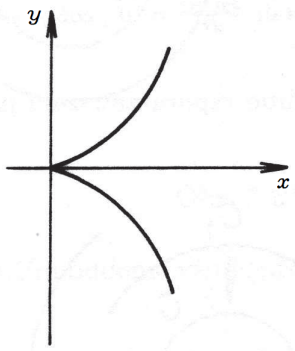
Označíme E_n jako Euklidovský n-rozměrný prostor. Tento prostor je isomorfní s prostorem $\mathbb{R}^n$. Pokud se budeme pohybovat v E_2, pak x, y budou kartézské souřadnice tohoto prostoru.

V následujícím textu budeme definovat pojem křivky. Zavedeme zobrazení $f : I \to E_n$, které se skládá z n složek reálných funkcí v kartézské soustavě souřadnic prostoru E_n, tj. $f(t) = (f^1(t), \ldots, f^n(t))$, kde $f^i : I \to \mathbb{R}$. $t \in I$ můžeme chápat jako čas a zobrazení f pak jako pohyb po trajektrii křivky.

Definice 1.2.1. Zobrazení $f : I \to E_n$ se nazývá *pohyb* v prostoru E_n a vektor $f' = \frac{df}{dt} : I \to E_n$ se nazývá vektor *rychlosti* pohybu f.

Definice 1.2.2. Řekneme, že pohyb $f : I \to E_n$ je *regulární*, jestliže $\frac{df(t)}{dt} \neq \vec{0} = (0, \ldots, 0)$ pro každé $t \in I$, tj. má v každém bodě nenulovou rychlost. Pokud $\frac{df(t_0)}{dt} = \vec{0}$, pak se bod t_0 nazývá *singulárním bodem* pohybu f.

Regulární pohyb znamená, že v každém bodě musí být rychlost nenulová, tedy že v každém bodě dokážeme sestavit tečný vektor. Příkladem neregulárního pohybu je semikubická parabola, pro kterou platí $f(t) = (t^2, t^3)$ (viz obrázek 1.1). V bodě $t_0 = 0$ je rychlost $f'(t_0)$ nulová, tedy bod t_0 je singulárním bodem.



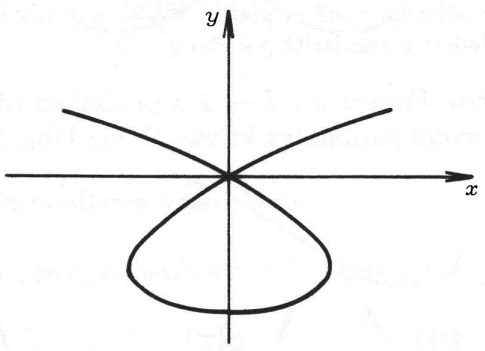
Obrázek 1.1: Příklad pohybu, který není regulární, zdroj [4]

Definice 1.2.3. Řekneme, že pohyb $f : I \to E_n$ je *jednoduchý pohyb*, jestliže platí:

$$t_1 \neq t_2 \Rightarrow f(t_1) \neq f(t_2).$$

Jedná se tedy o pohyb bez samoprotnutí.

Příkladem pohybu, který není jednoduchý, je pohyb znázorněný na obrázku 1.2. Jak později definujeme, vlastnost jednoduchého pohybu musí splňovat každá jednoduchá křivka. Pokud bychom připustili samoprotnutí pohybu, pak by v bodě protnutí vznikla nejednoznačnost v tečném vektoru (v tomto místě se nacházejí dva tečné vektory).



Obrázek 1.2: Příklad pohybu, který není jednoduchý, zdroj [4]

Poznámka 1.2.1. Pohyb je třídy C^r, jestliže všechny jeho souřadnice mají spojitou derivaci až do řádu r.

Definice 1.2.4. Množina $C \subset E_n$ se nazývá *jednoduchá křivka třídy* C^r, jestliže existuje takový jednoduchý regulární pohyb $f : I \to E_n$ třídy C^r, že platí $C = f(I)$. Zobrazení $f : I \to E_n$ se nazývá *parametrizace křivky* C.

Poznámka 1.2.2. Pokud je graf funkce $y = f(x)$, $x \in (a, b)$ třídy C^r jednoduchou křivkou C třídy C^r, pak pro tuto křivku vždy existuje parametrizace, která je rovna $g(t) = (t, f(t))$, protože platí, že $g'(t) = (1, f'(t)) \neq (0, 0)$.

Definice 1.2.5. Podmnožina $C \subset E_n$ se nazývá *křivka třídy* C^r, jestliže pro každý bod $X \in C$ existuje takové jeho okolí U v E_n, že $C \cap U$ je jednoduchá křivky třídy C^r. Parametrizace průniku $C \cap U$ nazýváme *lokální parametrizace křivky* C.

Definice 1.2.6. Nechť $f : I \to E_n$ je parametrizace jednoduché křivky $C = f(I)$. Tato parametrizace se nazývá *přirozená parametrizace*, jestliže platí

$$\left\| \frac{df}{ds} \right\| = 1 \quad pro \quad \forall s \in I,$$

kde s je přirozený parametr (oblouk).

Z fyzikálního hlediska se jedná o rovnoměrný pohyb po křivce, neboť při parametrizaci obloukem má pohyb po křivce v každém bodě jednotkovou rychlost.

Chceme-li najít přirozenou parametrizaci $g(s)$ k libovolné parametrizaci $f(t)$, přepočítáme parametr s tímto způsobem:

$$s = \int_{t_0}^{t} \sqrt{\left(\frac{df^1(\tau)}{d\tau}\right)^2 + \cdots + \left(\frac{df^n(\tau)}{d\tau}\right)^2}\, d\tau,$$

kde τ je původní parametr t, který přepíšeme pouze kvůli možnosti použít t v mezích integrálu. Jedná se o vzorec pro výpočet délky křivky. Tedy oblouk je parametr, který měří délku křivky.

Definice 1.2.7. Řekneme, že křivky C a $\bar{C}$ mají v bodě X *styk k-tého řádu*, jestliže

$$\frac{d^i f(s_0)}{ds^i} = \frac{d^i \bar{f}(s_0)}{ds^i} \quad \forall i = 1, \ldots, k.$$

Příkladem dvou křivek majících styk 1. řádu v bodě X je křivka C a její tečna v bodě X. Kromě tečny neexistuje jiná přímka, která by měla s křivkou styk 1. řádu. Mají-li proto dvě křivky C a $\bar{C}$ společnou tečnu v bodě X, pak v tomto bodě mají styk 1. řádu.

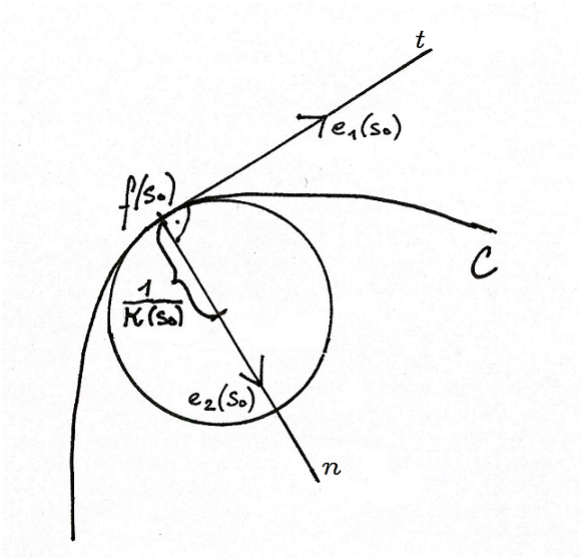
Definice 1.2.8. Bod $f(s_0) \in C$ nazýváme *inflexním bodem křivky* C, jestliže tečna v bodě $f(s_0)$ má s křivkou C styk 2. řádu.

Věta 1.2.1. *Bod $f(s_0)$ je inflexní právě tehdy, když $\frac{d^2 f(s_0)}{ds^2} = \vec{0}$.*

V dalším textu budeme uvažovat křivku C s přirozenou parametrizací $f : I \to E_2$. Nejprve zavedeme označení $e_1 = e_1(s) = \frac{df}{ds}$, kde e_1 je jednotkový tečný vektor. Vektor $\frac{de_1}{d_s}$ bude kolmý k vektoru e_1, což dokazuje následující. Jelikož je e_1 jednotkový vektor, pak platí $e_1 \cdot e_1 = 1$. Rovnici můžeme zderivovat podle s. Získáme tak $\frac{de_1}{ds} \cdot e_1 + e_1 \cdot \frac{de_1}{ds} = 0$, pak $2e_1 \cdot \frac{de_1}{ds} = 0$, pak $e_1 \cdot \frac{de_1}{ds} = 0$ a to znamená, že vektory jsou na sebe opravdu kolmé. Dále označíme $e_2(s_0)$ jako jednotkový vektor, který je kolmý na vektor $e_1(s_0)$ a stejně orientovaný jako vektor $\frac{de_1(s_0)}{ds}$. Jde tedy o jednotkový vektor normály a pro $\kappa(s_0) > 0$ platí

$$\frac{de_1(s_0)}{ds} = \kappa(s_0) \cdot e_2(s_0). \tag{1.1}$$

Věta 1.2.2. *V každém neinflexním bodě $f(s_0)$ křivky C existuje jediná kružnice, která má s křivkou C styk 2. řádu. Tato kružnice se nazývá oskulační kružnicí křivky C v bodě $f(s_0)$. Jejím středem je bod $f(s_0) + \frac{1}{\kappa(s_0)} \cdot e_2(s_0)$ a její poloměr je roven $\frac{1}{\kappa(s_0)}$.*



Obrázek 1.3: Oskulační kružnice křivky C

Definice 1.2.9. Číslo $\kappa(s_0)$ nazýváme *křivostí křivky* v neinflexním bodě $f(s_0)$ a platí

$$\kappa(s_0) = \left\| \frac{de_1(s_0)}{ds} \right\| = \left\| \frac{d^2 f}{ds^2} \right\|.$$

V inflexním bodě $f(s_0)$ definujeme křivost $\kappa(s_0) = 0$.

Křivost κ tedy vyjadřuje odchylku křivky od přímky.

Věta 1.2.3. *Při libovolné parametrizaci $f(t) = (x(t), y(t))$ křivky C je její křivost κ dána*

$$\kappa = \frac{x'y'' - y'x''}{((x')^2 + (y')^2)^{\frac{3}{2}}}. \tag{1.2}$$

1.3. Aproximace funkcí

Aproximaci funkce $f(x)$ budeme značit $\varphi(x)$ a píšeme, že $\varphi(x) \approx f(x)$. Budeme se zabývat dvěma základními typy, interpolací a aproximací metodou nejmenších čtverců, přičemž se omezíme pouze na případ, kdy $\varphi(x)$ je polynom. Následující informace jsou čerpány z [3],[9] a [12].

Interpolace polynomem

Interpolace je speciální případ aproximace, kde se funkce f a interpolace φ v jistých bodech shodují ve funkčních hodnotách a případně i v jejich derivacích.

Nechť jsou dány navzájem různé body

$$x_0, x_1, \ldots, x_n, \quad x_i \neq x_j \quad pro \quad i \neq j, \tag{1.3}$$

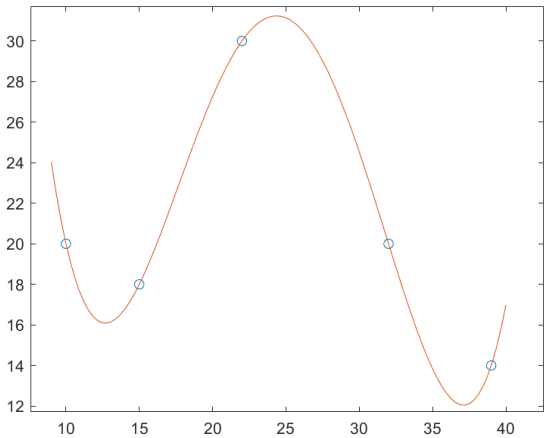
které nazveme *uzly*. Nechť v každém z těchto uzlů je definována funkční hodnota y_i. Budeme hledat interpolační polynom $P_n(x)$ stupně nejvýše n takový, že splňuje interpolační podmínky

$$P_n(x_i) = y_i, \quad i = 0, 1, \ldots, n. \tag{1.4}$$

Máme tedy $n + 1$ uzlových bodů, které dosadíme do rovnice polynomu ve tvaru

$$P_n(x) = a_0 + a_1 x + \cdots + a_n x^n \tag{1.5}$$

a získáme soustavu $n + 1$ lineárních rovnic o $n + 1$ neznámých koeficientů $a_k \in \mathbb{R}$, kde $k = 0, 1, \ldots, n$. Jejím vyřešením získáme hledanou interpolaci $\varphi(x)$ funkce $f(x)$.



Obrázek 1.4: Interpolace polynomem 4. stupně

Poznámka 1.3.1. V předchozím textu jsme se zabývali interpolačním polynomem, který má procházet zadanými hodnotami $P_n(x_i) = y_i$ v uzlech x_i. Pokud navíc požadujeme, aby hledaný polynom měl v uzlech předepsané derivace, hovoříme o *Hermitově interpolaci.*

Aproximace metodou nejmenších čtverců

V případě, že v uzlových bodech x_i neznáme přesné funkční hodnoty (například protože jsou zjištěny měřením, které nemusí být přesné), je nežádoucí, aby prokládaná křivka procházela přímo těmito body. Proto se použije aproximace, která nutně nemusí body funkce $f(x)$ procházet.

Mějme navzájem různé uzlové body $x_1, x_2, \ldots, x_n$ a příslušné funkční hodnoty $y_1, y_2, \ldots, y_n$ funkce $f(x)$, kterou však neznáme a pro kterou platí

$$y_i \approx f(x_i), \quad i = 0, 1, \ldots, n. \tag{1.6}$$

Nechť aproximační funkce φ má tvar

$$\varphi(\beta, x) = \beta_1 g_1(x) + \beta_2 g_2(x) + \cdots + \beta_k g_k(x), \tag{1.7}$$

kde k je počet jednotlivých bázových funkcí $g_i(x)$, které jsou lineárně nezávislé a musí platit $k \leq n$. $\beta = (\beta_1, \ldots, \beta_k)$ jsou neznámé koeficienty, které je třeba určit tak, aby kvadrát odchylek $\Phi(\beta)$ jednotlivých uzlů od aproximační funkce $\varphi(\beta, x)$ byl minimální. Hledáme tedy

$$\min_{\beta} \Phi(\beta) = \min_{\beta} \sum_{i=1}^{n} (\varphi(\beta, x_i) - y_i)^2. \tag{1.8}$$

Pro extrém funkce v bodě musí platit

$$\operatorname{grad} \Phi(\beta) = \mathbf{0}, \tag{1.9}$$

tedy

$$\begin{aligned}
\frac{\partial \Phi(\beta)}{\partial(\beta_1)} &= \sum_{i=1}^{n} 2(\varphi(\beta, x_i) - y_i)\beta_1 = 0 \\
\frac{\partial \Phi(\beta)}{\partial(\beta_2)} &= \sum_{i=1}^{n} 2(\varphi(\beta, x_i) - y_i)\beta_2 = 0 \\
&\vdots \\
\frac{\partial \Phi(\beta)}{\partial(\beta_k)} &= \sum_{i=1}^{n} 2(\varphi(\beta, x_i) - y_i)\beta_k = 0.
\end{aligned} \tag{1.10}$$

Úpravou získáme systém k lineárních rovnic o k neznámých

$$\begin{aligned}
\sum_{i=1}^{n} \varphi(\beta_1, \beta_2, \ldots, \beta_k, x_i) g_1(x_i) &= \sum_{i=1}^{n} y_i g_1(x_i) \\
\sum_{i=1}^{n} \varphi(\beta_1, \beta_2, \ldots, \beta_k, x_i) g_2(x_i) &= \sum_{i=1}^{n} y_i g_2(x_i) \\
&\vdots \\
\sum_{i=1}^{n} \varphi(\beta_1, \beta_2, \ldots, \beta_k, x_i) g_k(x_i) &= \sum_{i=1}^{n} y_i g_k(x_i).
\end{aligned} \tag{1.11}$$

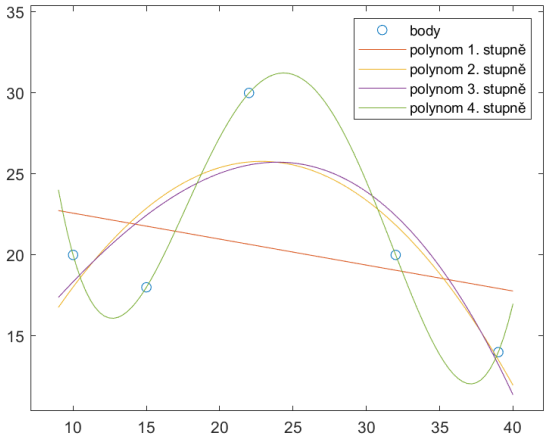
Rovnice lze přepsat do maticového zápisu, kde $g_j(x_i) = g_{i,j}$ pro $j = 1, \ldots, k$,

$$\begin{bmatrix} \sum_{i=1}^{n} g_{i,1}^2 & \sum_{i=1}^{n} g_{i,2}g_{i,1} & \cdots & \sum_{i=1}^{n} g_{i,k}g_{i,1} \\ \sum_{i=1}^{n} g_{i,1}g_{i,2} & \sum_{i=1}^{n} g_{i,2}^2 & \cdots & \sum_{i=1}^{n} g_{i,k}g_{i,2} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} g_{i,1}g_{i,k} & \sum_{i=1}^{n} g_{i,2}g_{i,k} & \cdots & \sum_{i=1}^{n} g_{i,k}^2 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i g_{i,1} \\ \sum_{i=1}^{n} y_i g_{i,2} \\ \vdots \\ \sum_{i=1}^{n} y_i g_{i,k} \end{bmatrix}, \tag{1.12}$$

neboli

$$\mathbf{G}\mathbf{G}^T\boldsymbol{\beta} = \mathbf{G}\mathbf{y}, \tag{1.13}$$

kde $\mathbf{G} = \begin{bmatrix} g_1(x_1) & \ldots & g_1(x_n) \\ \vdots & \ddots & \vdots \\ g_k(x_1) & \ldots & g_k(x_n) \end{bmatrix}$ je Gramova matice, $\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix}$.
Označme $\mathbf{H} = \mathbf{G}\mathbf{G}^T$. Matice $\mathbf{H}$ je symetrická a pozitivně semidefinitní, což je postačující podmínkou 2. řádu k tomu, že nalezený stacionární bod x_i je bod lokálního minima.



Obrázek 1.5: Aproximace metodou nejmenších čtverců

1.4. Fourierova transformace a konvoluce

Informace obsažené v této kapitole jsou čerpány z [7] a [8].

Libovolnou neperiodickou funkci, která má konečnou plochu pod křivkou, lze vyjádřit jako součet sinů a kosinů vynásobených váhovou funkcí. Tento součet se nazývá *Fourierova tranformace*. Jelikož se lze inverzním procesem vrátit k původní funkci bez jakékoliv ztráty informace, je Fourierova transformace vhodným nástrojem pro práci s obrazovou informací.

Definice 1.4.1. Označme $\mathcal{L}(\mathbb{R}^2)$ prostor všech funkcí $\mathbb{R}^2 \to \mathbb{C}$ takových, že integrál

$$\iint_{\mathbb{R}^2} |f(x,y)|\,dx\,dy \tag{1.14}$$

existuje a je konečný.

Pro další text budeme využívat rovnosti dle *Eulerova vzorce* $e^{i\varphi} = \cos(\varphi) + i\sin(\varphi)$, kde $i^2 = -1$.

Definice 1.4.2. Nechť $f(x,y) \in \mathcal{L}(\mathbb{R}^2)$. *Fourierova transformace* funkce f je funkce $\mathcal{F}\{f\}(\xi,\eta) = F(\xi,\eta) : \mathbb{R}^2 \to \mathbb{C}$ definována jako

$$F(\xi,\eta) = \iint_{\mathbb{R}^2} f(x,y)e^{-i(x\xi+y\eta)}\,dx\,dy. \tag{1.15}$$

Funkce F se také nazývá *Fourierovým spektrem* funkce f.

Definice 1.4.3. Nechť $G(\xi,\eta) \in \mathcal{L}(\mathbb{R}^2)$. *Inverzní Fourierova transformace* funkce G je funkce $\mathcal{F}^{-1}\{G\}(x,y) = g(x,y) : \mathbb{R}^2 \to \mathbb{C}$ definována jako

$$g(x,y) = \frac{1}{4\pi^2}\iint_{\mathbb{R}^2} G(\xi,\eta)e^{i(x\xi+y\eta)}\,d\xi\,d\eta. \tag{1.16}$$

Věta 1.4.1. ***(Fourierova inverzní věta pro funkci z $\mathcal{L}(\mathbb{R}^2)$)*** *Jestliže funkce $f(\xi,\eta), F(\xi,\eta) \in \mathcal{L}(\mathbb{R}^2)$ a f je spojitá funkce na $\mathbb{R}^2$, potom pro každé $(\xi,\eta) \in \mathbb{R}^2$ platí*

$$\mathcal{F}^{-1}\{\mathcal{F}\{f(x,y)\}\} = \frac{1}{4\pi^2}\iint_{\mathbb{R}^2} F(\xi,\eta)e^{i(x\xi+y\eta)}\,d\xi\,d\eta. \tag{1.17}$$

Definice 1.4.4. Nechť $f(x,y) \in \mathcal{L}(\mathbb{R}^2)$ má Fourierovo spektrum $F(\xi,\eta)$. *Amplitudové spektrum* funkce f je funkce $A(\xi,\eta) : \mathbf{R}^2 \to \mathbf{R}_0^+$ definovaná jako

$$A(\xi,\eta) = |\mathcal{F}\{f(x,y)\}| = |F(\xi,\eta)|. \tag{1.18}$$

Fázové spektrum funkce f je funkce $\phi(\xi,\eta) : \mathbb{R}^2 \to \langle 0, 2\pi)$ definovaná jako

$$\begin{aligned} \mathfrak{Re}F(\xi,\eta) &= A(\xi,\eta)\cos\phi(\xi,\eta), \\ \mathfrak{Im}F(\xi,\eta) &= A(\xi,\eta)\sin\phi(\xi,\eta). \end{aligned} \tag{1.19}$$

Definice 1.4.5. Nechť funkce $f_1(x,y), f_2(x,y) \in \mathcal{L}(\mathbb{R}^2)$. *Konvoluce* $f_1 * f_2$ funkcí f_1, f_2 je dána vztahem

$$f_1(x,y) * f_2(x,y) = \iint_{\mathbb{R}^2} f_1(s,t)f_2(x-s,y-t)\,ds\,dt. \tag{1.20}$$

Operace konvoluce má následující užitečné vlastnosti:

1. komutativnost:
$$f_1 * f_2 = f_2 * f_1 \tag{1.21}$$

2. násobení konstantou:
$$c_1 f_1 * c_2 f_2 = c_1 c_2 (f_1 * f_2) \tag{1.22}$$

3. distributivnost vůči sčítání:
$$f_1 * (f_2 + f_3) = f_1 * f_2 + f_1 * f_3 \tag{1.23}$$

4. asociativnost:
$$f_1 * (f_2 * f_3) = (f_1 * f_2) * f_3 \tag{1.24}$$

Věta 1.4.2. ***(Konvoluční teorém)*** *Nechť funkce* $f_1(x,y), f_2(x,y) \in \mathcal{L}(\mathbb{R}^2)$ *mají Fourierova spektra* $F_1(\xi,\eta), F_2(\xi,\eta)$. *Potom platí*

$$\mathcal{F}\{f_1(x,y) * f_2(x,y)\} = F_1(\xi,\eta) \cdot F_2(\xi,\eta). \tag{1.25}$$

Konvoluční teorém má velké využití při zpracování obrazové informace. Díky němu lze snadno přecházet mezi spektrální a prostorovou oblastí, což se využívá například u filtrace obrazu. Běžná filtrace ve spektrální oblasti totiž znamená vynásobení spektra filtrovaného obrazu frekvenční charakteristikou filtru (Fourierovým spektrem masky filtru).

Dosud jsme zvažovali funkci $f(x,y)$ definovanou jako $f : \mathbb{R}^2 \to \mathbb{C}$. Pokud ale hovoříme o obraze, můžeme předpokládat, že je konečný a definiční obor funkce f definovat jako ohraničenou oblast $\{0,1,\ldots,M-1\} \times \{0,1,\ldots,N-1\}$, kde $M, N \in \mathbb{N}$. Pak můžeme zavést *diskrétní Fourierovu transformaci* a *diskrétní inverzní Fourierovu transformaci* jako

$$\begin{aligned} \mathcal{D}\{f\}(\xi,\eta) = F_D(\xi,\eta) &= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) e^{-2\pi i(\frac{x\xi}{M}+\frac{y\eta}{N})}, \\ \mathcal{D}^{-1}\{f\}(x,y) &= \frac{1}{MN}\sum_{\xi=0}^{M-1}\sum_{\eta=0}^{N-1} F_D(\xi,\eta) e^{2\pi i(\frac{x\xi}{M}+\frac{y\eta}{N})}. \end{aligned} \tag{1.26}$$

Ve skutečnosti je ale obrazová funkce nekonečná a periodická a my jsme diskrétní Fourierovu transformaci a její inverzi definovali vztahy 1.26 pouze pro jednu periodu. Jelikož rovnice 1.26 platí pro každé $(\xi,\eta),(x,y) \in \mathbb{Z}^2$, můžeme pro obrazovou funkci definovat následující.

Definice 1.4.6. Nechť $f(x,y)$ je funkce $\{0,1,\ldots,M-1\} \times \{0,1,\ldots,N-1\} \to \mathbb{C}$, kde $M, N \in \mathbb{N}$ a nechť $F_D(\xi,\eta)$ je její Fourierovo spektrum. *Periodizací Fourierova spektra* F_D je funkce $\tilde{F}_D(\xi,\eta) : \mathbb{Z}^2 \to \mathbb{C}$ definovaná jako

$$\tilde{F}_D(\xi,\eta) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) e^{-2\pi i(\frac{x\xi}{M}+\frac{y\eta}{N})}. \tag{1.27}$$

Periodizací funkce f je funkce $\tilde{f}(x,y) : \mathbb{Z}^2 \to \mathbb{C}$ definovaná jako

$$\tilde{f}(x,y) = \frac{1}{MN}\sum_{\xi=0}^{M-1}\sum_{\eta=0}^{N-1} F_D(\xi,\eta) e^{2\pi i(\frac{x\xi}{M}+\frac{y\eta}{N})}. \tag{1.28}$$

Podobně jako ve spojitém případě, zavedeme konvoluci a konvoluční teorém pro diskrétní případ. Všechny vlastnosti konvoluce platí i zde.

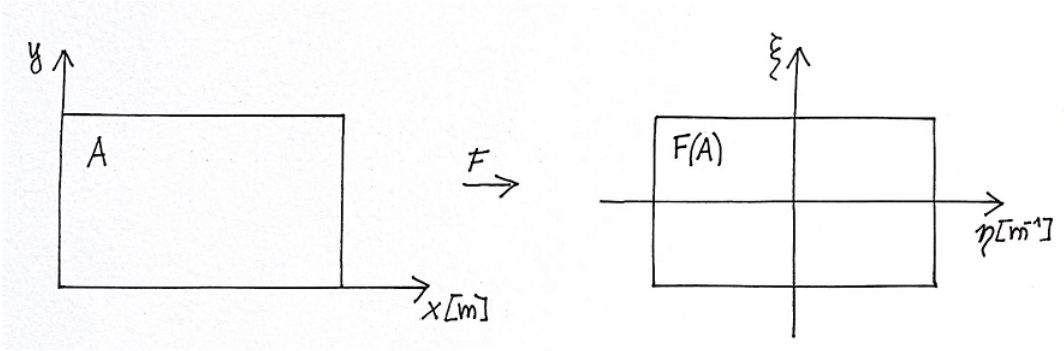
Definice 1.4.7. Nechť $f_1(x,y), f_2(x,y)$ jsou funkce $\{0,1,\ldots,M-1\}\times\{0,1,\ldots,N-1\} \to \mathbb{C}$, $M, N \in \mathbb{N}$. *Diskrétní periodická konvoluce* $f_1 * f_2$ funkcí f_1, f_2 je dána vztahem

$$f_1(x,y) * f_2(x,y) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f_1(s,t)\tilde{f}_2(x-s,y-t). \tag{1.29}$$

Věta 1.4.3. *Nechť funkce* $f_1(x,y), f_2(x,y) : \{0,1,\ldots,M-1\} \times \{0,1,\ldots,N-1\} \to \mathbb{C}$, $M, N \in \mathbb{N}$ *mají Fourierova spektra* $F_{D1}(\xi,\eta), F_{D2}(\xi,\eta)$. *Potom platí*

$$\mathcal{D}\{f_1(x,y) * f_2(x,y)\} = F_{D1}(\xi,\eta) \cdot F_{D2}(\xi,\eta). \tag{1.30}$$

Jelikož každý člen $F(\xi, \eta)$ obsahuje všechny hodnoty funkce $f(x, y)$ násobené exponenciálním členem, vztah mezi prostorovými charakteristikami obrazu a frekvenčními složkami Fourierovy transformace není možné viditelně pozorovat (viz obrázek 1.6). Několik zákonitostí zde ale obecně platí. Předpokládejme obraz v odstínech šedi. Protože frekvence je závislá na rychlosti změny intenzity pixelů v obraze, nulové frekvence ($\xi = \eta = 0$) odpovídají šedým plochám. Nízké frekvence odpovídají pomalu měnícím složkám obrazu a vysoké frekvence pak rychlejším změnám. Jedná se o hranice objektů a složky obrazu, které se vyznačují náhlými změnami úrovně jasu, jako je například šum.



Obrázek 1.6: Fourierova transformace obrazu A

Poznámka 1.4.1. Gaussova funkce se střední hodnotou $\mu = 0$ a směrodatnou odchylkou $\sigma = 1$ je funkce, která je sama sobě Fourierovou transformací. Nechť G_σ je Gaussova funkce se střední hodnotou μ a směrodatnou odchylkou σ, pak pro Fourierovu transformaci G_σ platí $\mathcal{F}(G_\sigma) = G_{\frac{1}{\sigma}}$.

2. Statistika

Pro posouzení kvality výsledků experimentu, kde porovnáváme vypočtené hodnoty s reálnými hodnotami, slouží dvě důležité kapitoly statistiky, korelační a regresní analýza. Korelační analýzou ověřujeme závislost náhodných proměnných a pokud existuje, pak za pomoci regresní analýzy vytváříme vhodný matematický model závislosti.

Informace k této kapitole jsou čerpány z [1], [2], [6].

2.1. Korelační analýza

Základní aplikace korelační analýzy je *výběrový korelační koeficient*, který vyjadřuje míru lineární stochastické závislosti mezi náhodnými proměnnými X a Y. Jedná se o výběr z nějakého dvojrozměrného rozdělení. Toto se dá zobecnit na $(k+1)$-rozměrné rozdělení, kde $\mathbf{X} = (X_1, \ldots, X_k)^T$ je náhodný vektor a Y je náhodná proměnná. Pak *výběrový koeficient mnohonásobné korelace* vyjadřuje míru závislosti mezi náhodnou proměnnou Y a nejlepší lineární kombinací složek náhodného vektoru $\mathbf{X}$. Speciální případ mnohonásobné korelace je *výběrový koeficient parciální korelace*. Tento koeficient se používá pro $(k+1)$-rozměrného rozdělení v případech, kdy je potřeba zjistit závislost mezi náhodnou proměnnou Y a konkrétní složkou náhodného vektoru X_i. Omezení vlivu ostatních složek vektoru $\mathbf{X}$ se provede jejich zkonstatněním.

Pro účely této práce se budeme dále věnovat pouze korelační analýze v dvourozměrném rozdělení, tedy výběrovému korelačnímu koeficientu.

Výše zmíněnou lineární závislost mezi náhodnými proměnnými X a Y vyjadřuje kovariance

$$C(X,Y) = E((X - EX)(Y - EY)),$$

kde EX, EY jsou střední hodnoty proměnných a korelace

$$\rho(X,Y) = \frac{C(X,Y)}{S(X)S(Y)}$$

se směrodatnými odchylkami $S(X) > 0$, $S(Y) > 0$ pro kterou platí $-1 \leq \rho(X,Y) \leq 1$. Pokud $\rho(X,Y) = 0$, pak proměnné X a Y jsou nezávislé. Pokud $\rho(X,Y) = 1$ (resp. -1), pak závislost $Y = a + bX$, $b > 0$ (resp. $b < 0$) platí s pravděpodobností rovné jedna.

Nechť $\begin{pmatrix} X_1 \\ Y_1 \end{pmatrix}, \ldots, \begin{pmatrix} X_n \\ Y_n \end{pmatrix}$ je náhodný výběr z vektoru $\begin{pmatrix} X \\ Y \end{pmatrix}$. Náhodné vektory $\begin{pmatrix} X_i \\ Y_i \end{pmatrix}$, $i \in \{1, \ldots, n\}$ jsou náhodným výběrem, jestliže jsou jednotlivé složky nezávislé a jestliže mají stejnou distribuční funkci. Pak definujeme výběrové statistiky:

1. výběrový průměr:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i, \quad \bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i \tag{2.1}$$

2. výběrový rozptyl:

$$S_X^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2, \quad S_Y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y})^2 \tag{2.2}$$

3. výběrová směrodatný odchylka:

$$S_X = \sqrt{S_X^2}, \quad S_Y = \sqrt{S_Y^2} \tag{2.3}$$

4. výběrový koeficient kovariance:

$$S_{XY} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y}) \tag{2.4}$$

5. výběrový koeficient korelace (Parsonův korelační koeficient):

$$r = \frac{S_{XY}}{\sqrt{S_X^2 S_Y^2}} = \frac{\sum_{i=1}^{n} X_i Y_i - n\bar{X}\bar{Y}}{\sqrt{\left(\sum_{i=1}^{n} X_i^2 - n\bar{X}^2\right)\left(\sum_{i=1}^{n} Y_i^2 - n\bar{Y}^2\right)}} \tag{2.5}$$

Jestliže je náhodný výběr z rozdělení, který má střední hodnoty μ_X, μ_Y, konečné rozptyly σ_X^2, σ_Y^2, kovarianci σ_{XY} a korelaci ρ, pak nejlepšími nestrannými odhady těchto parametrů jsou:

$$E(\bar{X}) = \mu_X, \quad E(\bar{Y}) = \mu_Y, \quad E(S_X^2) = \sigma_X^2 \quad E(S_Y^2) = \sigma_Y^2, \quad E(S_{XY}) = \sigma_{XY}.$$

Výběrový koeficient korelace r však není nestranným odhadem parametru ρ. Předpokládejme dvojrozměrné normální rozdělení $N_2(\mu_X, \mu_Y, \sigma_X^2, \sigma_Y^2, \rho)$ s kladnými rozptyly a korelačním koeficientem $\rho \in (-1, 1)$, pak pro ρ platí

$$E(r) = \rho - \frac{1-\rho^2}{n} + o(n^{-1}), \quad D(r) = \frac{(1-\rho^2)^2}{n} + o(n^{-1}).$$

Pro normální rozdělení s nulovým korelačním koeficientem, tj. $N_2(\mu_X, \mu_Y, \sigma_X^2, \sigma_Y^2, 0)$ a za předpokladu $n \geq 3$ dostaneme

$$E(r) = 0, \quad D(r) = \frac{1}{n-1}.$$

Věta 2.1.1. *Nechť* $\begin{pmatrix} X_1 \\ Y_1 \end{pmatrix}, \ldots, \begin{pmatrix} X_n \\ Y_n \end{pmatrix}$ *je náhodný výběr z dvojrozměrného normálního rozdělení, který má kladné rozptyly a korelační koeficient* $\rho = 0$. *Nechť* $n \geq 3$. *Pak*

$$t = \frac{r}{\sqrt{1-r^2}}\sqrt{n-2} \tag{2.6}$$

má rozdělení t_{n-2}, *tj. Studentovo rozdělení s* $(n-2)$ *stupni volnosti.*

Poznámka 2.1.1. Chceme-li testovat nezávislost proměnných X a Y, tedy testovat, zda jejich koeficient korelace je roven 0 (tj. $\rho(X, Y) = 0$), pak využijeme předchozí věty 2.1.1.

Výběrový korealční koeficient (Pearsonův korelační koeficient)

Nechť $\begin{pmatrix} X_1 \\ Y_1 \end{pmatrix}, \ldots, \begin{pmatrix} X_n \\ Y_n \end{pmatrix}$ je náhodný výběr z dvojrozměrného normálního rozdělení, počet pokusů $n \geq 3$ a koeficient korelace $\rho = 0$. Testujeme hypotézu $H_0 : \rho = 0$ proti alternativní hypotéze $H_A : \rho \neq 0$. Jestliže testovací kritérium

$$t = \frac{r}{\sqrt{1 - r^2}} \sqrt{n - 2},$$

kde r je výběrový korelační koeficient z 2.5, náleží doplňku kritického oboru

$$\bar{W}_\alpha = \Big\langle -t_{n-2}(1 - \alpha/2), t_{n-2}(1 - \alpha/2) \Big\rangle,$$

pak hypotézu H_0 nezamítáme na hladině významnosti α. Tzn. nezamítáme hypotézu, že jsou proměnné X a Y nezávislé.

V obecném případě, kdy korelační koeficient není nulový, není vhodné použít předchozí statistiku. V takových případech se výběrový korelační koeficient transformuje *Fisherovou z-transformací* (více informací např. v [2]).

V praxi však často neznáme rozdělení pravděpodobnosti a tedy nemůžeme splnit předpoklad normálního rozdělení dat. Proto je vhodné testovat nezávislost *Spearmanovým korelačním koeficientem*, který porovnává pořadí náhodných proměnných X a Y.

Spearmanův korelační koeficient

Nechť $\begin{pmatrix} X_1 \\ Y_1 \end{pmatrix}, \ldots, \begin{pmatrix} X_n \\ Y_n \end{pmatrix}$ je náhodný výběr ze spojitého dvojrozměrného rozdělení. Nechť hodnoty $R_1, \ldots, R_n$ jsou odpovídající pořadí náhodné proměnné $X_1, \ldots, X_n$ a hodnoty $Q_1, \ldots, Q_n$ jsou pořadí proměnné $Y_1, \ldots, Y_n$.
Testujeme hypotézu $H_0 : \rho = 0$ proti alternativní hypotéze $H_A : \rho \neq 0$ a testovacím kritériem je

$$r_S = 1 - \frac{6}{n(n^2 - 1)} \sum_{i=1}^{n} (R_i - Q_i)^2.$$

V případě, že $n \leq 30$, jsou kritické hodnoty $r_S(\alpha)$ uvedeny v tabulce na obrázku 2.1.

	α			α			α	
n	0,05	0,01	n	0,05	0,01	n	0,05	0,01
			11	0,6091	0,7545	21	0,4351	0,5545
			12	0,5804	0,7273	22	0,4241	0,5426
			13	0,5549	0,6978	23	0,4150	0,5306
			14	0,5341	0,6747	24	0,4061	0,5200
5	0,9000	–	15	0,5179	0,6536	25	0,3977	0,5100
6	0,8286	0,9429	16	0,5000	0,6324	26	0,3894	0,5002
7	0,7450	0,8929	17	0,4853	0,6152	27	0,3822	0,4915
8	0,6905	0,8571	18	0,4716	0,5975	28	0,3749	0,4828
9	0,6833	0,8167	19	0,4579	0,5825	29	0,3685	0,4744
10	0,6364	0,7818	20	0,4451	0,5684	30	0,3620	0,4665

Obrázek 2.1: Kritické hodnoty $r_S(\alpha)$ pro Spearmanův korelační koeficient, zdroj [2]

Pro $n > 30$ se pro kritickou hodnotu využije asymptotická normalita koeficientu r_S, která se vypočte jako

$$r_S^*(\alpha) = \frac{u(\alpha/2)}{\sqrt{n-1}},$$

kde $u(\alpha/2)$ je je kvantil normovaného normálního rozdělení.

Hypotézu H_0, tj. hypotézu o nezávislosti náhodných proměnných X a Y zamítáme na hladině významnosti α, jestliže platí $|r_S| \geq r_S(\alpha)$ (případně $|r_S| \geq r_S^*(\alpha)$).

2.2. Regresní analýza

Mějme náhodný vektor $\boldsymbol{X} = (X_1, \ldots, X_k)$ nezávisle proměnných a závisle proměnnou Y. Závislost mezi nimi vyjadřuje regresní funkce

$$y = \varphi(\boldsymbol{x}, \boldsymbol{\beta}) + e, \tag{2.7}$$

kde vektor $\boldsymbol{x} = (x_1, \ldots, x_k)$ jsou hodnoty náhodného vektoru $\boldsymbol{X}$, y je hodnota závislé proměnné Y, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)$ je vektor hledaných *regresních koeficientů* a e je nezávislá náhodná proměnná s rozdělením pravděpodobnosti $N(0, \sigma^2)$, která vyjadřuje chybu.

V první řadě musíme stanovit tvar regresní funkce, což provedeme na základě grafického znázornění provedených pokusů n. Tato funkce se bude skládat z tzv. *bázových funkcí* g_j, které musí být lineárně nezávislé a regresních koeficientů β_j. Úkolem regresní analýzy je odhadnout hodnoty koeficientů β_j. Navržené regresní funkce (modely) se dělí na lineární a nelineární, kde linearita znamená, že β_j není argumentem bázových funkcí, tedy model je lineární vzhledem ke koeficientům β_j. Například i regresní model $y_i = \beta_1 x_i + \beta_2 sin(x_i)$ je lineární, naopak model $y_i = \beta_1 x_i + \beta_2 sin(\beta_3 x_i)$ lineární není. V dalším textu se budeme zabývat pouze lineárním regresním modelem, jehož funkce má tvar

$$y_i = \sum_{j=1}^{k} \beta_j g_j(\boldsymbol{x}_i) + e_i, \quad i = 1, \ldots, n. \tag{2.8}$$

Nejčastěji používanými lineárními regresními funkcemi jsou:

- regresní přímka: $y_i = \beta_0 + \beta_1 x_i$
- regresní parabola: $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$

Odhady neznámých regresních koeficientů β_j, $j = 1, \ldots, k$ získáme minimalizováním tzv. *reziduálního součtu čtverců*

$$S^* = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{k} \beta_j g_{ji} \right)^2, \tag{2.9}$$

který odpovídá metodě nejmenších čtverců.

Regresní model

Mějme n pokusů, pak získáme $(k+1)$-rozměrný statistický soubor $((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n))$ s rozsahem n. Symbolem $\boldsymbol{X}$ nyní označíme náhodný výběr z náhodného vektoru, tedy $\boldsymbol{X} = (x_{ij})$ je matice typu n x k, kde $k < n$. Náhodný výběr z proměnné Y označíme jako $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T$ a $\boldsymbol{e} = (e_1, \ldots, e_n)^T$ bude vektor nezávislých náhodných proměnných s normálním rozdělením pravděpodobnosti pro které pro $\forall i, j = 1, \ldots, n$, $i \neq j$ platí

$$E(e_i) = 0, \quad D(e_i) = \sigma^2 > 0, \quad C(e_i, e_j) = 0.$$

Lineární regresní model tedy můžeme zapsat jako

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{e} \tag{2.10}$$

a předpokládáme, že matice $\boldsymbol{X}$ má lineárně nezávislé sloupce a platí pro ni, že $k < n$, tzn. $\boldsymbol{X}^T\boldsymbol{X}$ je pozitivně definitní matice. Vektor $\boldsymbol{X}\boldsymbol{\beta}$ je nenáhodný a proto platí

$$\begin{aligned} &E(\boldsymbol{Y}) = E(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{e}) = E(\boldsymbol{X}\boldsymbol{\beta}) + E(\boldsymbol{e}) = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{0} = \boldsymbol{X}\boldsymbol{\beta}, \\ &var(\boldsymbol{Y}) = var(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{e}) = var(\boldsymbol{e}) = \sigma^2\boldsymbol{I}. \end{aligned}$$

Reziduální součet čtverců viz 2.9, který chceme minimalizovat, můžeme zapsat jako výraz

$$(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) \tag{2.11}$$

a odhady koeficientů $\boldsymbol{\beta}$ označíme jako $\boldsymbol{b} = (b_1, \ldots, b_k)$. Pak bodový odhad $\boldsymbol{b}$ metodou nejmenších čtverců je roven

$$\boldsymbol{b} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}. \tag{2.12}$$

Poznámka 2.2.1. Soustava lineárních rovnic $\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{b} = \boldsymbol{X}^T\boldsymbol{Y}$ z které vyplývá odhad koeficientů $\boldsymbol{\beta}$, se nazývá *soustava normálních rovnic*.

Poznámka 2.2.2. Matice $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$ se nazývá projekční matice. Je symetrická, idempotentní a její stopa se rovná $h(\boldsymbol{H}) = k$. Použijeme-li toto označení, pak vektor $\hat{\boldsymbol{Y}} = \boldsymbol{X}\boldsymbol{b} = \boldsymbol{H}\boldsymbol{Y}$ je nejlepší aproximací vektoru $\boldsymbol{Y}$, která se dá vytvořit lineární kombinací sloupců matice $\boldsymbol{X}$.

Poznámka 2.2.3. Náhodná proměnná

$$s^2 = \frac{S_e}{n-k}, \tag{2.13}$$

kde $S_e = S^*_{min} = (\boldsymbol{Y} - \hat{\boldsymbol{Y}})^T(\boldsymbol{Y} - \hat{\boldsymbol{Y}})$, je bodový odhad rozptylu σ^2.

Regresní přímka

Pro praktickou část práce zde vyjádříme konkrétní vztahy pro regresní model obecné přímky, tj.

$$y_i = \beta_0 + \beta_1 x_i + e_i, \quad i = 1, \ldots, n.$$

Bodový odhad regresních koeficientů $\boldsymbol{\beta} = (\beta_0, \beta_1)$ získáme metodou nejmenších čtverců. Platí

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ .... & \\ 1 & x_n \end{pmatrix}, \quad \boldsymbol{X}^T\boldsymbol{X} = \begin{pmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{pmatrix}, \quad \boldsymbol{X}^T\boldsymbol{Y} = \begin{pmatrix} \sum y_i \\ \sum x_i y_i \end{pmatrix}.$$

Označme

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i,$$

pak odhady koeficientů $\boldsymbol{\beta}$ jsou

$$b_1 = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2}, \quad b_0 = \bar{y} - b_1\bar{x} \tag{2.14}$$

a odhad rozptylu σ^2 je

$$s^2 = \frac{\sum_{i=1}^{n} y_i^2 - b_0\sum_{i=1}^{n} y_i - b_1\sum_{i=1}^{n} x_i y_i}{n-2}. \tag{2.15}$$

Vhodnost regresního modelu

Výsledný reziduální součet $S_e = (\boldsymbol{Y} - \hat{\boldsymbol{Y}})^T(\boldsymbol{Y} - \hat{\boldsymbol{Y}})$ vyjadřuje reziduální variabilitu regresního modelu, tj. odchylku měřených hodnot od vypočtených hodnot. Celková variabilita vysvětluje odchylku měřených hodnot od průměru a je rovna $S_T = \sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n} y_i^2 - n\bar{y}^2$. Přesnost navrženého modelu popisuje *koeficient determinace*, který vychází z těchto variabilit. Nabývá hodnot z intervalu $\langle 0, 1\rangle$ a je roven

$$R^2 = 1 - \frac{S_e}{S_T}. \tag{2.16}$$

Koeficient po převodu na procenta udává, kolik procent bodů je vysvětleno navrženým modelem. Tedy čím větší je hodnota R^2, tím těsnější je regresní závislost.

Test hypotézy

Nechť $\boldsymbol{V} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}$, kde v_{jj}, $j = 1, \ldots, k$ je diagonální prvek matice $\boldsymbol{V}$.
Testujeme hypotézu $H_0 : \beta_j = \beta_j^0$ proti alternativní hypotéze $H_A : \beta_j \neq \beta_j^0$. Jestliže testovací kritérium

$$t = \frac{b_j - \beta_j^0}{\sqrt{s^2 v_{jj}}},$$

kde s^2 je bodový odhad rozptylu z 2.13, náleží doplňku kritického oboru

$$\bar{W}_\alpha = \langle -t_{n-k}(1-\alpha/2), t_{n-k}(1-\alpha/2)\rangle,$$

pak hypotézu H_0 nezamítáme na hladině významnosti α. t_{n-k} je kvantil Studentova rozdělení s $(n-k)$ stupni volnosti.

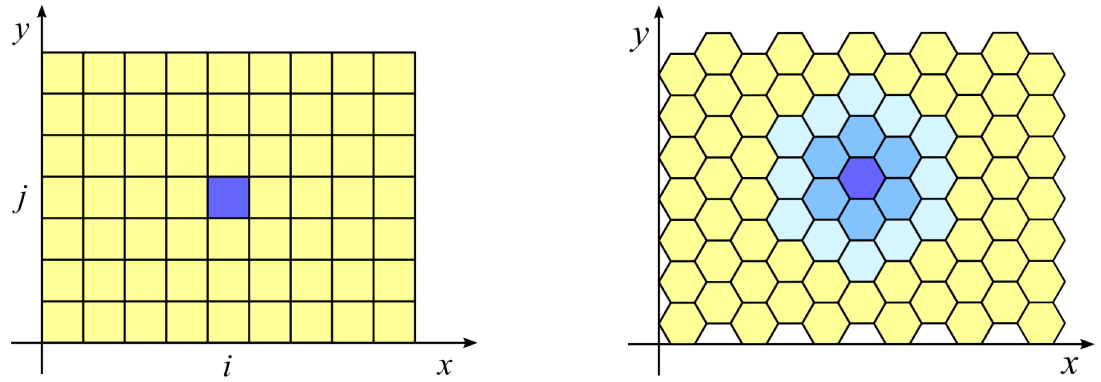
Poznámka 2.2.4. Nejčastější aplikací tohoto testu je testovaní koeficientu β_j na hodnotu $\beta_j^0 = 0$, který ověří významnost koeficientu v modelu.

3. Zpracování obrazové informace

Zpracování obrazové informace se dělí na několik částí, které se liší oblastí, v níž se aplikuje. V tomto textu se budeme zabývat dvěma z těchto částí. Prvně se jedná o image restoration, neboli rekonstrukci obrazu, kde je hlavním cílem potlačení šumu v obraze a obnovení ztracených informací při jeho snímání. Druhou částí je image analysis, neboli analýza obrazu. Ta se zabývá především získáváním určitých vlastností obsažených v obrazové informaci. Následující informace jsou čerpány z [7], [8], [10] a [12].

3.1. Vlastnosti obrazu

Obrazem rozumíme *obrazovou matici* o rozměrech $m \times n$, kterou budeme značit $\mathbf{I}$, resp. $\mathbf{I}_{m\times n}$. Prvek obrazové matice se na nazývá *obrazový element* nebo *pixel*. Poloha pixelu je určena řádkovým indexem j a sloupcovým indexem i (viz obrázek 3.1a). Pixel $\mathcal{P}$ o souřadnicích $[i, j]$ obrazové matice $\mathbf{I}_{m\times n}$, kde $0 \leq i \leq m-1$, $0 \leq j \leq n-1$ budeme značit $\mathcal{P}[i, j]$. Obrazová matice může mít obecně libovolnou geometrii (obrázek 3.1). Nejčastěji používána je však *pravoúhlá matice*, kde pixely mají tvar čtverce nebo obdélníku. Dále v textu budeme uvažovat vždy pravoúhlou obrazovou matici.



(a) pravoúhlá (b) hexagonální

Obrázek 3.1: Obrazová matice, zdroj [12]

Definice 3.1.1. Mějme obrazovou matici $\mathbf{I}_{m\times n} = (\mathcal{P}[i, j])$ a konstantu $r > 0$. Množinu pixelů $\mathcal{Q}[k, l]$, pro kterou platí

$$\sqrt{(k-i)^2 + (l-j)^2} \leq r \tag{3.1}$$

nazveme *okolím pixelu* $\mathcal{P}[i, j]$ a budeme ho značit $\mathcal{N}^{\mathcal{P}}(i, j, r)$.

3.1.1. Atributy pixelu

Pixel $\mathcal{P}$ je většinou definován nějakou n-ticí hodnot. Například u barevného obrazu se jedná o trojici hodnot vyjadřující intenzity červené, zelené a modré složky světla. Atributy pixelu $\mathcal{P}$ jsou jeho vlastnosti, které vyplývají právě z intenzit složek světla. Dělí se na vlastní a nevlastní. Nevlastní atributy potřebují k určení i okolí daného pixelu.

Vlastní atributy pixelu

Tyto atributy vycházejí pouze z hodnot daného pixelu. Budeme uvažovat barevný obraz, tzn. jeden pixel bude reprezentován třemi hodnotami. Bude to intenzita jasu červené barvy - R, zelené barvy - G a modré barvy - B, kde $R, G, B \in \langle 0, w \rangle$. w je maximální hodnota pro daný dynamický rozsah obrazu, konkrétně pro 8-bitový obraz je $w = 255$. Základní vlastní atributy pixelu jsou:

1. intenzity složek barvy červené, zelené a modré: R, G, B
2. jas pixelu J:
$$J = \frac{R + G + B}{3}, \quad J \in \langle 0, w \rangle \tag{3.2}$$
3. hustota pixelu d:
$$d = \frac{J}{w}, \quad d \in \langle 0, 1 \rangle \tag{3.3}$$
4. saturace barvy S:
$$S = 1 - \frac{\min\{R, G, B\}}{\max\{R, G, B\}}, \quad \max\{R, G, B\} > 0, \quad S \in \langle 0, 1 \rangle \tag{3.4}$$
5. světlost pixelu L:
$$L = \frac{\max\{R, G, B\} + \min\{R, G, B\}}{2}, \quad L \in \langle 0, w \rangle \tag{3.5}$$
6. tón barvy H: udává tzv. úhel barvy. Pro čistě červenou $H = 0$, čistě zelenou $H = \frac{2}{3}\pi$ a čistě modrou $H = \frac{4}{3}\pi$.
$$\begin{aligned} max &= \max\{R, G, B\} \\ min &= \min\{R, G, B\} \\ H &= \frac{\pi}{3}\frac{G - B}{max - min} \quad \text{pro} \quad max = R \wedge G \geq B \\ H &= 2\pi + \frac{\pi}{3}\frac{G - B}{max - min} \quad \text{pro} \quad max = R \wedge G < B \\ H &= \frac{2\pi}{3} + \frac{\pi}{3}\frac{B - R}{max - min} \quad \text{pro} \quad max = G \\ H &= \frac{4\pi}{3} + \frac{\pi}{3}\frac{R - G}{max - min} \quad \text{pro} \quad max = B \\ H &\text{ není definováno pro } max = 0. \end{aligned} \tag{3.6}$$

V případě *černobílého obrazu* (tj. obraz ve stupních šedi), platí $R = G = B = J = L \in \langle 0, w \rangle$, saturace barvy je $S = 0$ a tón barvy H není definován. Pak pixely v obrazové matici jsou reprezentovány právě jednou hodnotou. Speciálním případem je *binární obraz*, kde pixely nabývají pouze hodnot 0 (černá) nebo w (bílá).

Nevlastní atributy pixelu

Pro uvedené nevlastní atributy budeme předpokládat reprezentaci pixelu $\mathcal{P}$ pouze jednou hodnotou, což není nijak omezující pro použití, jelikož se v praxi pracuje s jednotlivými složkami obrazu zvlášť.

Nevlastní atributy pixelu jsou zejména závislé na vlastních atributech sousedních pixelů. Dále jsou také závislé na velikosti a tvaru uvažovaného okolí. Pro daný pixel $\mathcal{P} = \mathcal{P}[i,j]$ jsou nevlastní atributy:

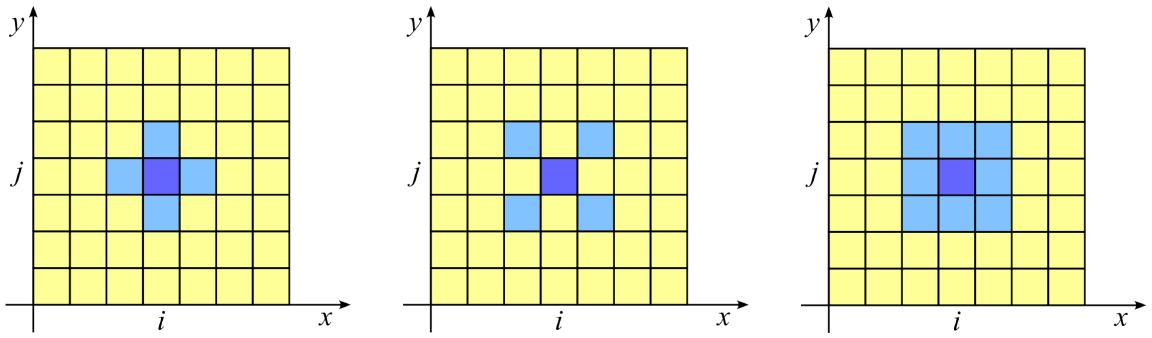
1. střední hodnota: $\text{Mean}^{\mathcal{P}}(i,j) = \frac{1}{n}\sum_{k=1}^{n}\mathcal{P}^k$, kde $\mathcal{P}^k$ jsou pixely z nějakého okolí pixelu $\mathcal{N}^{\mathcal{P}}(i,j)$ a n je počet pixelů v daném okolí
2. rozptyl hodnot: $\text{Var}^{\mathcal{P}}(i,j) = \frac{1}{n}\sum_{k=1}^{n}[\mathcal{P}^k - \text{Mean}^{\mathcal{P}}(i,j)]^2$, kde $\mathcal{P}^k$ jsou pixely z okolí pixelu $\mathcal{N}^{\mathcal{P}}(i,j)$ a n je počet pixelů v daném okolí
3. gradient: $\text{Grad}^{\mathcal{P}}(i,j) = \sqrt{(\mathcal{P}[i+1,j] - \mathcal{P}[i,j])^2 + (\mathcal{P}[i,j+1] - \mathcal{P}[i,j])^2}$. Z matematického hlediska se jedná o velikost gradientu.

3.1.2. Základní vazby pixelů

Pro popis metod analýzy obrazu je nutné definovat základní vazby mezi jednotlivými pixely v obraze.

Okolí pixelu

V předchozím textu jsme si již zavedli okolí pixelu $\mathcal{N}^{\mathcal{P}}(i,j,r)$. Následně však definujeme i jeho podmnožiny pro vhodná $r > 0$, které se používají v praktických i teoretických úlohách. Nejpoužívanější typy okolí pixelu $\mathcal{P}[i,j]$ jsou tzv. *čtyřsousednost* $\mathcal{N}_4^{\mathcal{P}}(i,j)$, *diagonální sousednost* $\mathcal{N}_{\mathcal{D}}^{\mathcal{P}}(i,j)$ a *osmisousednost* $\mathcal{N}_8^{\mathcal{P}}(i,j)$, které jsou znázorněny na obrázku 3.2.



(a) $\mathcal{N}_4^{\mathcal{P}}(i,j)$ (b) $\mathcal{N}_{\mathcal{D}}^{\mathcal{P}}(i,j)$ (c) $\mathcal{N}_8^{\mathcal{P}}(i,j)$

Obrázek 3.2: Základní typy okolí pixelu, zdroj [12]

Spojitelnost

Spojitelnost je vlastnost pixelů, která je velmi užitečná při určování hranice objektu nebo nějaké oblasti. Jestliže máme rozhodnout, zda jsou dva pixely spojitelné, pak musí být k sobě přináležející (sousedící ve smyslu definovaného okolí) a jejich hodnoty musí být nějak podobné, např. si musí být rovny.

Definice 3.1.2. Mějme dva pixely $\mathcal{P}[i,j]$ a $\mathcal{Q}[k,l]$. Jestliže $\mathcal{Q}[k,l] \in \mathcal{N}_4^{\mathcal{P}}(i,j)$ řekneme, že pixely $\mathcal{P}[i,j]$ a $\mathcal{Q}[k,l]$ jsou *4-spojitelné*. Jestliže $\mathcal{Q}[k,l] \in \mathcal{N}_8^{\mathcal{P}}(i,j)$ řekneme, že pixely $\mathcal{P}[i,j]$ a $\mathcal{Q}[k,l]$ jsou *8-spojitelné*. Pixely $\mathcal{P}[i,j]$ a $\mathcal{Q}[k,l]$ jsou *spojitelné* resp. sousedící, pokud tyto pixely jsou 4-spojitelné nebo 8-spojitelné.

Definice 3.1.3. Mějme množinu $\mathcal{O}$ a dva různé pixely $\mathcal{P}_0[x_0,y_0]$ a $\mathcal{P}_n[x_n,y_n]$. Nechť existuje posloupnost $\mathcal{P}_k[x_k,y_k] \in \mathcal{O}$ vzájemně různých pixelů, kde $k = 0,...,n$. Pokud jsou pixely $\mathcal{P}_k[x_k,y_k]$ a $\mathcal{P}_{k+1}[x_{k+1},y_{k+1}]$, kde $k = 0,...,n-1$, spojitelné, pak tuto posloupnost pixelů $\mathcal{P}_k[x_k,y_k]$ nazveme *cestou*.

3.2. Prvky v obraze

Tato kapitola definuje důležité pojmy prvků, které se vyskytují v obraze a které budeme v dalším textu využívat.

Definice 3.2.1. Nechť T_k, $k = 1,\ldots,n$ jsou atributy pixelů a n je jejich počet a nechť atribut T_k může nabývat hodnot $t_k \in \langle t_k^l, t_k^h\rangle$, kde t_k^l je minimální a t_k^h je maximální hodnota pro daný atribut. Množinu $T = \{T_1, T_2, \ldots, T_n\}$ nazýváme *atributovou množinou*.

Definice 3.2.2. Mějme intervaly $\langle \tau_k^l, \tau_k^h\rangle \subseteq \langle t_k^l, t_k^h\rangle$, kde $k = 1,\ldots,n$ a nechť t_k jsou hodnoty atributu T_k daného pixelu $\mathcal{P}$. Pixel $\mathcal{P}$, který splňuje podmínku $t_k \in \langle \tau_k^l, \tau_k^h\rangle$, kde $k = 1,\ldots,n$, nazveme *objektovým pixelem* a řekneme, že pixel $\mathcal{P}$ má vlastnost τ. V opačném případě $\mathcal{P}$ nazveme *pixelem pozadí*.

Volba atribuční množiny T a intervalů $\langle \tau_k^l, \tau_k^h\rangle$ je důležitá pro správnou identifikaci objektových pixelů. Tento krok není obecně jednoznačný a velmi závisí na vlastnostech celkového obrazu a objektu, který máme identifikovat.

3.2.1. Objekt

Definice 3.2.3. Nechť $\mathcal{O}$ je množina pixelů a pro všechny dvojice pixelů $\mathcal{P}_0, \mathcal{P}_n \in \mathcal{O}$ existuje alespoň jedna cesta jako posloupnost bodů $\mathcal{P}_k \in \mathcal{O}$, kde $k = 0,...,n$ z bodu $\mathcal{P}_0$ do bodu $\mathcal{P}_n$, pak tuto množinu nazveme *souvislou množinou* nebo též *objektem*.

Nyní je důležité si uvědomit, že typ spojitelnosti má velký vliv při identifikaci objektů. Může se lišit výsledný tvar, velikost i počet jednotlivých objektů. Proto je podstatné nejprve zvážit, jaký typ spojitelnosti použijeme.

Definice 3.2.4. Mějme objekt $\mathcal{O}$ a okolí $\mathcal{N}_4^{\mathcal{P}}$ objektového pixelu $\mathcal{P} \in \mathcal{O}$. *Hraničním pixelem* nazveme pixel $\mathcal{P}$, jestliže alespoň jeden pixel z okolí $\mathcal{N}_4^{\mathcal{P}}$ je pixel pozadí. Množinu všech hraničních pixelů objektu $\mathcal{O}$ budeme značit $\partial\mathcal{O}$.

Poznámka 3.2.1. Hraniční pixely definované pomocí $\mathcal{N}_4^{\mathcal{P}}$ okolí jsou 8-spojitelné. Jestliže definujeme hraniční pixely pomocí $\mathcal{N}_8^{\mathcal{P}}$ okolí, pak jsou 4-spojitelné.

Definice 3.2.5. Mějme obrazovou matici $\mathbf{I}_{m\times n}$ a objekt $\mathcal{O}$. Jestliže množina pixelů $\mathbf{I}_{m\times n} \setminus \mathcal{O}$ bude tvořit jeden objekt a $\mathcal{O} \setminus \partial\mathcal{O}$ také jeden objekt, pak řekneme, že objekt $\mathcal{O}$ je *jednoduše souvislý*.

3.2.2. Digitální křivka

Definice 3.2.6. Nechť $\mathcal{O}$ je množina pixelů, $\mathcal{P} \in \mathcal{O}$ a pixel $\mathcal{P}$ je spojitelný maximálně s jedním pixelem, patřícím do množiny $\mathcal{O}$, pak pixel $\mathcal{P}$ nazveme *koncovým pixelem.*

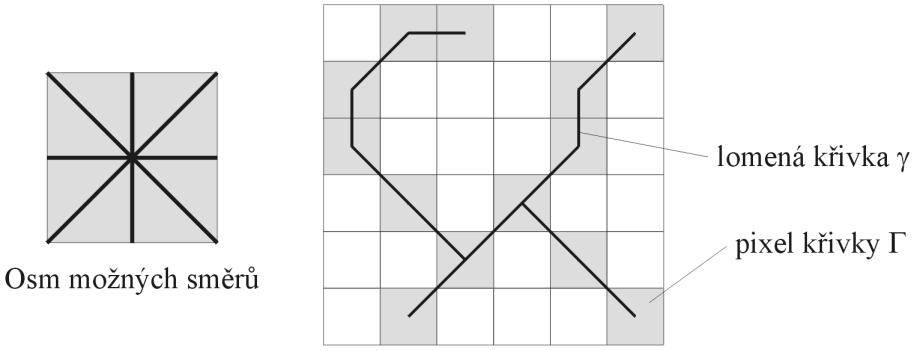
Definice 3.2.7. Mějme množinu pixelů Γ, která má určitý počet koncových pixelů. Jestliže po vyjmutí jakéhokoliv pixelu z množiny Γ, který není koncovým pixelem, se počet koncových pixelů zvýší, pak množinu Γ nazveme *digitální křivkou.*

Definice 3.2.8. Mějme množinu $\Gamma = \{\mathcal{P}_k\}$, kde $\mathcal{P}_k$, $k = 0, \ldots, n$, je posloupnost vzájemně různých pixelů. Nechť posloupnost pixelů $\mathcal{P}_k$ tvoří jedinou cestu z pixelu $\mathcal{P}_0$ do pixelu $\mathcal{P}_n$, pak množinu pixelů Γ nazveme *jednoduchou digitální křivkou.*

Definice 3.2.9. Mějme množinu $\Gamma = \{\mathcal{P}_k\}$, $k = 0, ..., n$. Nechť pixely $\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$ tvoří posloupnost vzájemně různých pixelů a $\mathcal{P}_n = \mathcal{P}_0$. Jestliže po vyjmutí jakéhokoliv pixelu z množiny Γ vznikne jednoduchá digitální křivka, pak množinu Γ nazveme *uzavřenou digitální křivkou.*

Pokud sjednotíme dvě digitální křivky, výsledkem nemusí být znovu digitální křivka. Pokud objektem $\mathcal{O}$ je digitální křivka, pak všechny pixely $\mathcal{P} \in \mathcal{O}$ jsou hraničními pixely objektu $\mathcal{O}$. Obrácená implikace obecně neplatí.

Délku libovolné digitální křivky Γ určíme pomocí křivky γ, která je po částech lineární a prochází středy jednotlivých pixelů (viz. obrázek 3.3). V těchto středech se může křivka γ lomit nebo větvit do maximálně osmi směrů. Délku části křivky γ pro daný pixel $\mathcal{P} \in \Gamma$ určíme tak, že sečteme délky všech úseček jednotlivých použitých směrů. Takto projdeme všechny pixely křivky Γ a po sečtení dílčích délek dostáváme délku digitální křivky Γ.



Obrázek 3.3: Lomená křivka γ, zdroj [12]

3.3. Filtrace obrazu

Obraz vyfocený reálným obrazovým detektorem není nikdy dokonalý a pokud s ním chceme dále pracovat, je vhodné nejprve využít některé filtrační techniky. Filtrací obrazu můžeme například odstranit nebo naopak zvýraznit jeho zrnitost, případně můžeme zvýraznit objekty v obraze.

Pod pojmem filtr budeme rozumět konvoluci jádra $\mathbf{H}$ s obrazovou maticí $\mathbf{I}$. Fourierovo spektrum jádra budeme nazývat frekvenční charakteristikou filtru. Dále budeme vždy uvažovat černobílý obraz.

3.3.1. Šum

Každý obraz je při snímání degradován náhodnými složkami *šumu*. V následujícím textu budou popsány dva možné typy vyskytujícího šumu v obraze, *aditivní šum* a *impulsní šum*.

Aditivní šum

V čipu fotoaparátu se vyskytují „bludné“ elektrony, které do fotografie vnáší šum. Jejich množství je závislé na teplotě a ochlazováním čipu můžeme jejich výskyt omezit. Tomuto jevu se říká *temný proud*. A právě temným proudem je převážně způsoben aditivní šum.

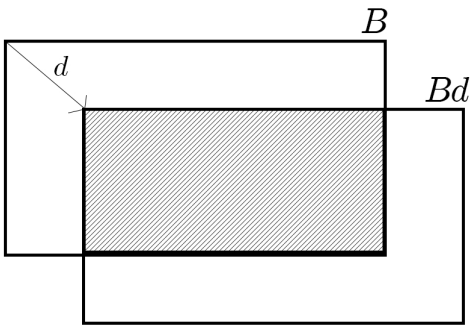
Definice 3.3.1. Označme A digitální černobílý ideální obraz (tzn. obraz, který neobsahuje šum). Nechť S je digitální černobílý obraz stejné velikosti jako obraz A, jehož hodnoty pixelů jsou stochasticky nezávislé realizace náhodné veličiny X. Nechť $B = A + S$. Potom obraz B obsahuje *aditivní šum*. Obraz S se nazývá *obraz šumu* a charakteristiky X nazýváme *charakteristiky aditivního šumu*.

Předchozí definice nezohledňuje případy $A+S<0$ a $A+S\geq w$, kde w je dynamický rozsah obrazu. Proto obraz B definujeme následně:

$$B(x,y)=\begin{cases} 0 & \text{pokud} \quad A(x,y)+S(x,y)<0 \\ A(x,y)+S(x,y) & \text{pokud} \quad 0<A(x,y)+S(x,y)<w \\ w-1 & \text{pokud} \quad A(x,y)+S(x,y)\geq w \end{cases} . \tag{3.7}$$

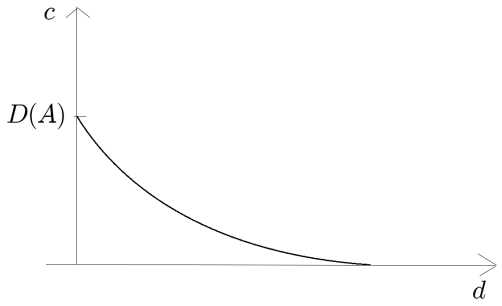
Obrazy šume S pořízené CCD a CMOS čipy mají y pravidla normální rozdělení $N(\mu,\sigma^2)$. Chceme-li určit množství aditivního šumu v obraze, vhodnou charakteristikou bude směrodatná odchylka σ. Čím větší je σ, tím horší je obraz, tj. obsahuje více šumu. (Střední hodnota μ vhodná není, protože závisí na expozičním čase. Čím delší je expoziční doba, tím více elektronů na čip dopadne a celý obraz je více světlejší. Střední hodnota je pak vyšší a ztrácíme část dynamického rozsahu obrazu.)

Předpokládejme, že obrazy A a S jsou nezávislé, pak můžeme pro odhad σ použít autokovariační funkci. Mějme obraz Bd, který je identický obrazu B. Autokovariační funkce těchto obrazů je pak $c(d)=C(B,Bd)$ (viz obrázek 3.4). Platí, že $c(d)$ s rostoucím d klesá k nule a pro d jdoucí k nule se $c(d)$ blíží k $D(B)$ (rozptylu obrazu B).



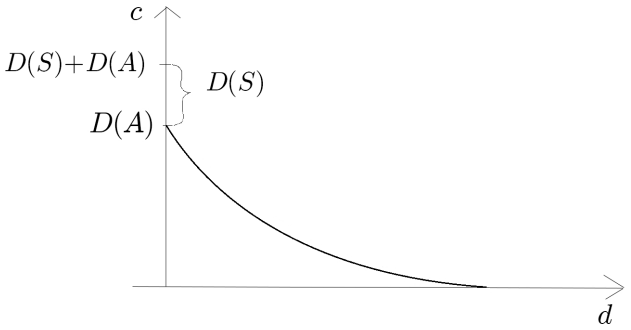
Obrázek 3.4: Vznik autokovariační funkce $c(d)$, zdroj [7]

Na obrázku 3.5 jsou znázorněny autokovariační funkce ideálního obrazu A, obrazu šumu S a obrazu $B = A + S$, z něhož lze odvodit míru aditivního šumu přibližně rovnou $\sigma = \sqrt{D(S)}$.



(a) ideální obraz A

(b) obraz šumu S



(c) obraz $B = A + S$

Obrázek 3.5: Autokovariační funkce, zdroj [7]

Impulsní šum

Impulsní šum je způsoben vadami na čipu, chybami při dálkovém přenosu dat a dopadajícími miony, což jsou částice pohybující se v kosmu. Při dlouhých expozicích, dopadne na čip více mionů a proto je impulsní šum závislý na expozičním čase. Dále je také závislý na nadmořské výšce.

Definice 3.3.2. Označme A digitální černobílý ideální obraz. Nechť B je digitální černobílý obraz splňující

$$B(x,y) = \begin{cases} A(x,y) & \text{pokud} \quad Y = 0 \\ X & \text{pokud} \quad Y = 1 \end{cases}, \tag{3.8}$$

kde Y je náhodná proměnná s Alternativním rozdělením pravděpodobnosti a X je náhodná proměnná s libovolným rozdělením pravděpodobnosti. Potom řekneme, že obraz B obsahuje *impulsní šum.*

Filtrace impulsního šumu spočívá nejprve v jeho detekci a poté v opravě nalezených vadných pixelů.

Detekce je založena na testovaní hypotézy, že daný pixel je vadný. Testovacím kritériem může být například kritérium

$$|v - \bar{v}| > \varepsilon, \tag{3.9}$$

kde v je hodnota pixelu, $\bar{v}$ je libovolná statistika z okolí pixelu (např. aritmetický průměr, medián nebo modus) a ε je konstanta, která ovlivňuje hladinu významnosti a sílu testu. Při testování může nastat chyba 1. druhu, tzn. pixel je vadný, ale test hypotézu zamítne (nenajdeme všechny vadné pixely) nebo chyba 2. druhu, tzn. pixel je správný, ale test hypotézu nezamítne (kazíme obraz). Chyba 2. druhu je pro výsledný obraz mnohem horší a proto je vhodné volit ε vyšší a zvyšovat tak spíš chybu 1. druhu.

Při opravě, neboli korekci vadných pixelů, opravujeme pouze pixely vybrané pomocí předchozí detekce. Existuje více metod jak postupovat. One-pass metoda nalezený vadný pixel rovnou opraví zvolenou statistikou $\bar{v}$. Double-pass metoda nejprve provede celou detekci a pak při korekci pixelu do statistiky $\bar{v}$ nezahrne jiné vadné pixely, které se nacházejí v okolí opravovaného pixelu. Interpolační metoda nejprve provede celou detekci a poté pro každý opravovaný pixel na základě správných pixelů v okolí spočítá funkci, kterou aproximuje obraz.

3.3.2. Lineární filtry

Lineární filtry jsou lineární ve smyslu platnosti principu superpozice nad vstupy a výstupy. Tyto filtry jsou vhodné zejména pro filtraci aditivního šumu. Upravují frekvenční spektrum zpracovávaného obrazu a zvolením konvolučního jádra docílíme jejich různých vlastností.

Definice 3.3.3. Mějme obrazové matice $\mathbf{I}^I = (\mathcal{P}[i,j])$, $\mathbf{I}^O = (\mathcal{Q}[i,j])$ a konvoluční jádro $\mathbf{H} = (h[k,l])$ o rozměrech $2m+1 \times 2n+1$, kde $k = 0, \ldots, 2m$ a $l = 0, \ldots, 2n$. Zobrazení $\mathcal{F}_{lin} : \mathcal{P}[i,j] \to \mathcal{Q}[i,j]$, které je dáno vztahem

$$\mathcal{Q}[i,j] = \sum_{k=-m}^{m} \sum_{l=-n}^{n} \mathcal{P}[i-k, j-l] \cdot h[m-k, n-l] \tag{3.10}$$

nazveme *lineárním filtrem.*

Základní typy lineárních filtrů a několik jejich konkrétních příkladů popisuje následující text.

Filtr typu identický obraz

Jádro *filtru typu identický obraz* obsahuje pouze Diracův impuls. Pro jádro velikosti 3×3 je tedy

$$\mathbf{H}_I = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{3.11}$$

Jelikož je součet všech prvků jádra roven 1, filtr nemění střední hodnotu obrazu. Tento filtr nepotlačuje žádné prostorové frekvence a po jeho aplikaci získáme znovu původní obraz.

Filtr typu dolní propust

Obecně platí, že *filtr typu dolní propust* zachovává nízké prostorové frekvence a upravuje vysoké prostorové frekvence. Skok z jedné jasové úrovně šedi na druhou přemění na postupný přechod mezi nimi. Tedy použitím tohoto filtru celý obraz rozostříme a potlačíme tak i aditivní šum v obraze.

Nejlepší konvoluční jádra pro filtr typu dolní propust jsou jádra vytvořená z Gaussovské funkce dvou proměnných G_σ, tedy například jádra

$$\mathbf{H}_L = \begin{bmatrix} \frac{1}{10} & \frac{1}{10} & \frac{1}{10} \\ \frac{1}{10} & \frac{1}{5} & \frac{1}{10} \\ \frac{1}{10} & \frac{1}{10} & \frac{1}{10} \end{bmatrix}, \mathbf{H}_L = \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix}, \tag{3.12}$$

kde součet všech hodnot prvků jádra je roven jedné.

Připomeňme, že pro Gaussovu funkci G_σ je Fourierova transformace rovna $\mathcal{F}(G_\sigma) = G_{\frac{1}{\sigma}}$. Čím větší směrodatnou odchylku σ použijeme v Gaussově funkci G_σ, tím menší směrodatnou odchylku bude mít její Fourierova transformace $\mathcal{F}(G_\sigma)$ a výsledný obraz po aplikaci takového konvolučního jádra bude více rozmazán. Platí tedy, že s vyšší volbou σ získáme rozmazanější obraz a vyfiltrujeme více šumu v obraze.

Filtr typu horní propust

Filtr typu horní propust naopak od dolní propusti propouští vysoké prostorové frekvence a ty nízké potlačuje. Využívá se ke zvýraznění vysokých frekvencí, což jsou především hrany vyskytující se v obraze.

Konvoluční jádro tohoto typu získáme odečtením filtrovaného obrazu filtrem typu dolní propust od původního obrazu. Označme $\mathbf{I}$ jako obrazovou matici původního obrazu, $\mathbf{H}_L = G_{\frac{1}{\sigma}}$ je jádro filtru typu doplní propust a $\mathbf{H}_I$ je jádro filtru typu identický obraz. Platí

$$\mathbf{I} - \mathbf{I} * \mathbf{H}_L = (\mathbf{I} * \mathbf{H}_I - \mathbf{I} * \mathbf{H}_L) = \mathbf{I} * (\mathbf{H}_I - \mathbf{H}_L) = \mathbf{I} * \mathbf{H}_H, \tag{3.13}$$

tedy jádro horní propusti je dáno jako $\mathbf{H}_H = \mathbf{H}_I - G_{\frac{1}{\sigma}}$. Příklady konvolučního jádra typu horní propust jsou

$$\mathbf{H}_H = \begin{bmatrix} -\frac{1}{10} & -\frac{1}{10} & -\frac{1}{10} \\ -\frac{1}{10} & \frac{4}{5} & -\frac{1}{10} \\ -\frac{1}{10} & -\frac{1}{10} & -\frac{1}{10} \end{bmatrix}, \mathbf{H}_H = \begin{bmatrix} -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \\ -\frac{1}{8} & \frac{3}{4} & -\frac{1}{8} \\ -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \end{bmatrix}. \tag{3.14}$$

3.3.3. Nelineární filtry

Rozdílem mezi lineárními a nelineárními filtry je, že lineární nahrazují pixely nějakými nově vypočtenými hodnotami intenzit. Nelineární filtry pouze vyberou vhodnou hodnotu intenzity z okolí daného pixelu, tedy nepřidávají do obrazu žádné nové hodnoty.

Dále uvedeme nejjednodušší typy z nelineárních filtrů. Jejich vlastnosti lze ovlivnit velikostí používaného okolí $\mathcal{N}^{\mathcal{P}}(i,j)$.

Mediánový filtr

Definice 3.3.4. Mějme obrazové matice $\mathbf{I}^I = (\mathcal{P}[i,j])$, $\mathbf{I}^O = (\mathcal{Q}[i,j])$ a okolí pixelu $\mathcal{N}^{\mathcal{P}}(i,j)$. Zobrazení $\mathcal{F}_{\text{med}} : \mathcal{P}[i,j] \to \mathcal{Q}[i,j]$, které je dáno vztahem

$$\mathcal{Q}[i,j] = \text{median}\,\mathcal{N}^{\mathcal{P}}(i,j) \tag{3.15}$$

nazveme *mediánovým filtrem.*

Tento filtr filtruje šum v obraze a zároveň zachovává hrany objektů, což je jeho velká výhoda. Sice nezpřesní informaci obsaženou v obraze, ale je dobrým nástrojem před další analýzou obrazu a proto je hojně využíván.

Filtr typu maximum a minimum

Definice 3.3.5. Mějme obrazové matice $\mathbf{I}^I = (\mathcal{P}[i,j])$, $\mathbf{I}^O = (\mathcal{Q}[i,j])$ a okolí pixelu $\mathcal{N}^{\mathcal{P}}(i,j)$. Zobrazení $\mathcal{F}_{\max} : \mathcal{P}[i,j] \to \mathcal{Q}[i,j]$, které je dáno vztahem

$$\mathcal{Q}[i,j] = \max \mathcal{N}^{\mathcal{P}}(i,j) \tag{3.16}$$

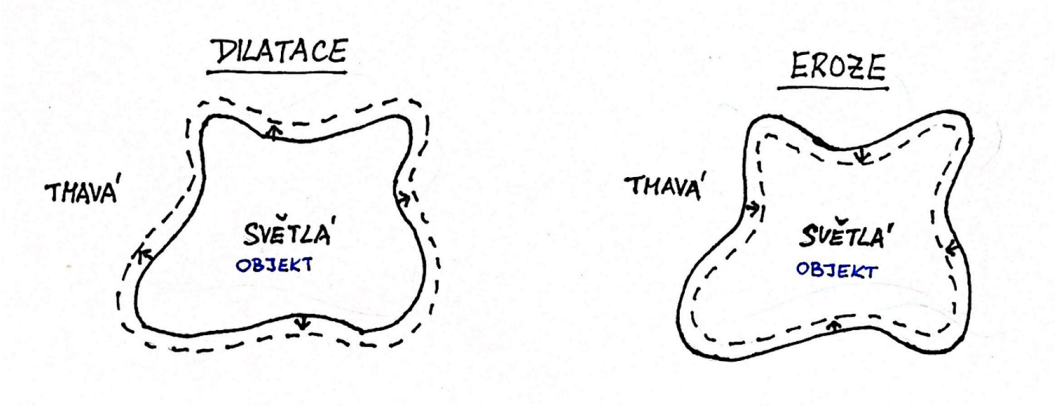
nazveme *filtrem typu maximum.*

Definice 3.3.6. Mějme obrazové matice $\mathbf{I}^I = (\mathcal{P}[i,j])$, $\mathbf{I}^O = (\mathcal{Q}[i,j])$ a okolí pixelu $\mathcal{N}^{\mathcal{P}}(i,j)$. Zobrazení $\mathcal{F}_{\min} : \mathcal{P}[i,j] \to \mathcal{Q}[i,j]$, které je dáno vztahem

$$\mathcal{Q}[i,j] = \min \mathcal{N}^{\mathcal{P}}(i,j) \tag{3.17}$$

nazveme *filtrem typu minimum.*

Mějme obraz, kde mezi objektem a pozadím je velký kontrast (není nutný čistě binární obraz). Hodnoty pixelů pozadí budou tmavé (hodnoty blížící se k 0) a objektové pixely budou světlé (hodnoty blížící se k 1). Pak filtr typu maximum nazveme *dilatací* a filtr typu minimum *erozí* (viz obrázek 3.6). V případě opačných hodnot objektových pixelů a pixelů pozadí, označíme filtry typu maximum a minimum přesně naopak. Avšak pojmy dilatace a eroze chápeme stále stejně (dilatace- rozšíření objektu, eroze- zmenšení objektu). Dále v textu budeme vždy předpokládat situaci uvedenou na obrázku 3.6.



Obrázek 3.6: Dilatace a eroze

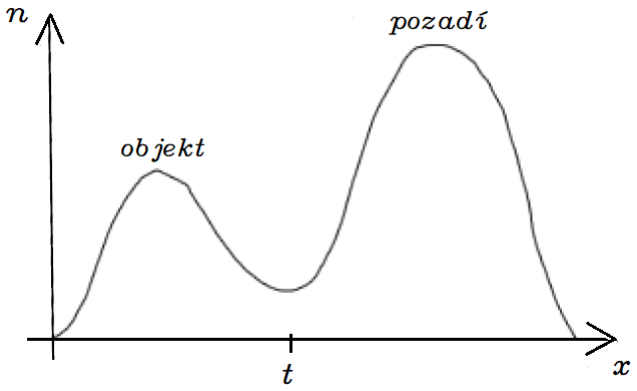
Poznámka 3.3.1. Dilataci a erozi lze použít i pro detekci hran v obraze. Konkrétně bude tento postup popsán v kapitole 3.4.2 Morfologické operace - Dilatace a Eroze.

3.4. Analýza obrazu - segmentace

Jak již bylo řečeno, analýza obrazu se zabývá získáváním konkrétní informace z obrazu. Pojem segmentace je proces, který vyčleňuje odlišné oblasti obrazu se stejnými rysy z okolních oblastí. Konkrétně rozděluje obraz na pixely pozadí a objektové pixely. K takovému rozdělení je třeba zvolit kritérium, podle kterého budeme obraz segmentovat.

3.4.1. Prahování

Metoda prahování je vhodná pouze, pokud má obraz silně bimodální histogram, tj. například histogram dle obrázku 3.7. Na tomto obrázku je znázorněn histogram obrazu, ve kterém se nachází tmavý objekt na světlém pozadí.



Obrázek 3.7: Bimodální histogram (tmavý objekt na světlém pozadí)

Prvně se z histogramu určí prahová hodnota t, tedy hodnota, která definuje přechod mezi objektem a pozadím. Poté se testuje hypotéza, že daný pixel $\mathcal{P}$ patří do objektu. Pro výše popsanou situaci (tmavý objekt na světlém pozadí) je testovací kritérium

$$x(\mathcal{P}) \leq t. \tag{3.18}$$

V opačném případě (světlý objekt na tmavém pozadí) je testovací kritérium ve tvaru $x(\mathcal{P}) \geq t$.

V praxi se často stává, že prahová hodnota t není stejná pro celý obraz. To je většinou způsobené nehomogením osvětlením. Vznikají viditelné stíny nebo například vinětace, což je jev, při kterém jsou okraje obrazu méně nasvícené a jsou tedy tmavší než střed obrazu. V takovém případě se musí obraz před samotným prahováním nejprve kalibrovat, nebo se musí využít jiná metoda segmentace.

Adaptivní prahování

Adaptivní prahování je takové prahování, kde práh t je závislý na souřadnicích pixelu $\mathcal{P}[i,j]$. Prahové hodnoty t_{ij} získáme z histogramu h_{ij} vytvořeného z okolí $\mathcal{N}^{\mathcal{P}}(i,j)$ pixelu $\mathcal{P}[i,j]$. Potom testujeme hypotézu, že pokud platí

$$x(\mathcal{P}[i,j]) \leq t_{ij}, \tag{3.19}$$

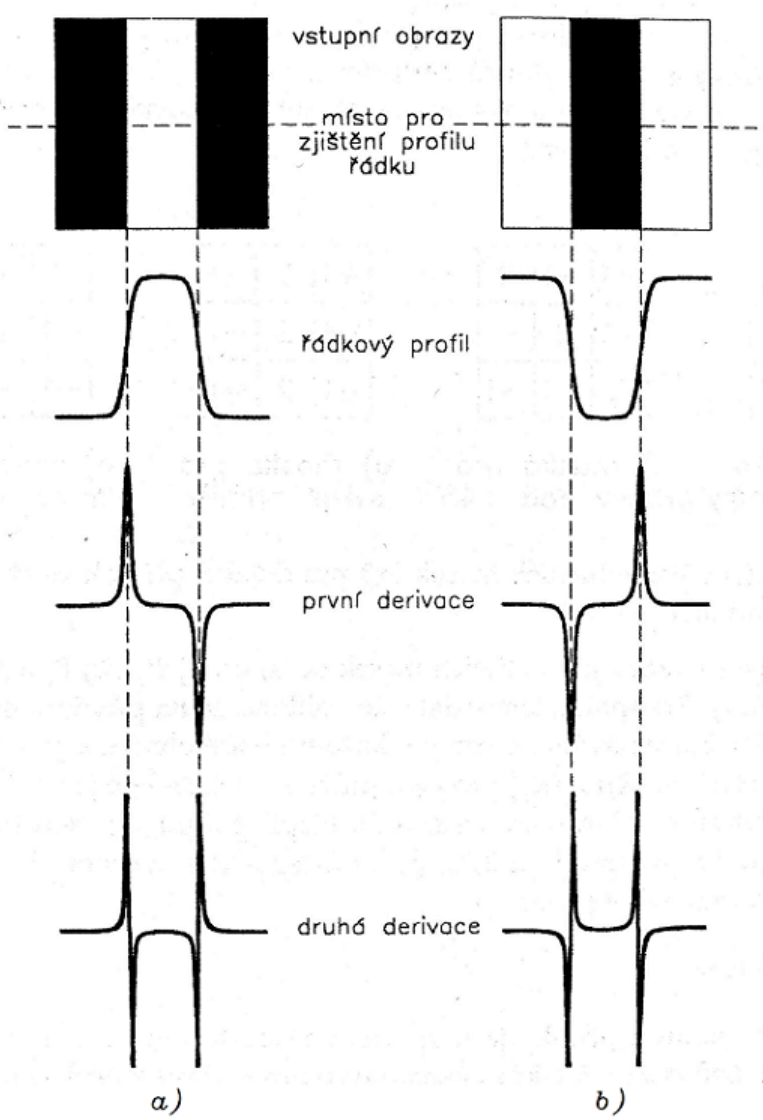
pak pixel $\mathcal{P}[i,j]$ je objektovým pixelem.

Každý histogram h_{ij} by měl být opět bimodální, tzn. že velikost okolí by měla být zvolena tak, aby se v každém vybíraném okolí $\mathcal{N}^{\mathcal{P}}(i,j)$ nacházely jak objektové pixely, tak i pixely pozadí. V případě, že tato podmínka nebude dodržena, vzniknou ve výsledném vysegmentovaném obraze nové „falešné“ objekty nebo naopak v objektu vzniknou „falešné“ díry. Tyto chybně označené pixely pak považujeme za impulsní šum a můžeme je filtrovat.

3.4.2. Detekce hran

Existuje mnoho metod určených k detekci hran. V následujícím textu si však uvedeme pouze některé z nich, konkrétně gradientní metody, metodu diskrétního Laplaciánu a použití dilatace a eroze.

Jedna velká skupina těchto metod je založená na první a druhé derivaci. Otázku, proč je derivace vhodná pro nalezení hranice objektu, vysvětluje obrázek 3.8. Mějme obraz, který obsahuje světlý pruh na tmavém pozadí (3.8a) a obraz, který obsahuje tmavý pruh na světlém pozadí (3.8b). Řádkový profil zobrazuje hodnoty pixelů v jakémkoliv horizontálním řádku. Pak z grafu první derivace vyplývá, že ji lze použít pro posouzení, zda se v obraze nachází hrana objektu. Znaménko druhé derivace určí, jaký jas má hraniční pixel, tj. jestli jsou jeho hodnoty tmavé či světlé. Dále můžeme pozorovat, že z grafu druhé derivace lze odečíst přesná poloha hranice, a to v místě, kde graf prochází nulou.



Obrázek 3.8: První a druhá derivace řádkového profilu dvou vzájemně inverzních obrazů, zdroj [8]

Podobným způsobem lze v obraze hledat hranu jakékoliv orientace. První derivace v jakémkoliv bodě obrazu je daná velikostí gradientu v tomto bodě. Druhá derivace je získána pomocí Laplaceova operátoru.

Gradientní metody

Už podle názvu je jasné, že gradientní metody jsou založeny na gradientu.

V praxi se využívají odvozená konvoluční jádra (např. uvedeme Prewittova a Sobelova), pomocí kterých spočteme první a druhou složku gradientu. Konvolucí obrazové matice a horizontální masky $\mathbf{H}_x$ získáme složku gradientu G_x a konvolucí s vertikální maskou $\mathbf{H}_y$ získáme složku gradientu G_y. Výsledkem je pak gradientní obraz, jehož velikost je stejná jako velikost původního obrazu a který obsahuje úrovně jasu $J \in \langle 0, w \rangle$. To však znamená, že na získaný gradientní obraz je pak nutné ještě aplikovat některou z metod prahování.

Prewittova konvoluční jádra mají tvar

$$\mathbf{H}_x^P = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \mathbf{H}_y^P = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{3.20}$$

a Sobelova konvoluční jádra jsou ve tvaru

$$\mathbf{H}_x^S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \mathbf{H}_y^S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}. \tag{3.21}$$

Jelikož derivace obecně zvýrazňují šum (vysoké frekvence) v obraze, je nutné před použitím hranových detektorů založených na derivacích původní obraz filtrovat.

Poznámka 3.4.1. Z výsledných složek gradientu lze spočítat síla hrany a to pomocí velkosti gradientu $I = \sqrt{G_x^2 + G_y^2}$.

Metoda diskrétního Laplaciánu

Podobně jako u gradientních metod se aplikuje i tato metoda. Rozdílná jsou pouze použitá konvoluční jádra, která jsou založena na druhé derivaci a mají tvar

$$\mathbf{H}_1^L = \begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix}, \mathbf{H}_2^L = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \tag{3.22}$$

Součet všech koeficientů Laplaceovy masky musí být roven nule, aby v případě homogenní plochy v obraze byla odezva nulová.

Jedná se vlastně o filtr typu horní propust, který zvýrazňuje vysoké frekvence v obraze, tedy hrany. Nevýhodou je, že ve vysokých frekvencích se vyskytuje i šum. Stejně jako u gradientních metod je tedy tato metoda závislá na množství šumu v obraze.

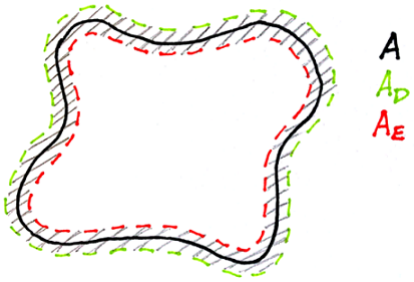
Morfologické operace - Dilatace a Eroze

V kapitole 3.3.3 jsme zavedli pojmy dilatace a eroze. Pomocí těchto operací lze nalézt hranice objektů v obraze následujícím způsobem.

Nechť A je obraz obsahující objekt a pozadí s velkým kontrastem mezi nimi. Obraz A_D je obraz vytvořený pomocí dilatace a obraz A_E pomocí eroze. Pak

$$B = A_D - A_E \tag{3.23}$$

je obraz, který obsahuje hranici objektu. Celý tento postup znázorňuje obrázek 3.9, kde šrafovaná část je výsledný obraz B.

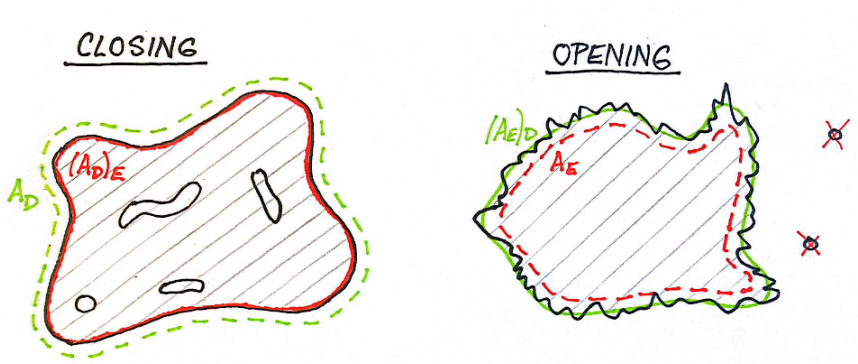


Obrázek 3.9: Detekce hran dilatací a erozí

Získanou oblast, v níž se hranice nachází, lze následně zúžit pomocí eroze. Pokud je našim cílem získat digitální křivku hranice, pouhá eroze ale nestačí. Výsledný objekt by nejen zužovala, ale také zkracovala, což je nežádoucí. Proto se v praxi spíše využívá tzv. *podmíněná eroze*, která zohledňuje, jestli se jedná o pixel zakončující křivku nebo pixel, který by způsobil rozpad objektu. Následně u takových pixelů erozi neprovede.

Poznámka 3.4.2. V případě, že erozi, resp. podmíněnou erozi provádíme pomocí okolí $\mathcal{N}_4(i,j)$, výsledkem budou hraniční pixely objektu, které jsou 8-spojitelné. V případě použitého okolí $\mathcal{N}_8(i,j)$ budou hraniční pixely 4-spojitelné.

Dalším využitím dilatace a eroze je tzv. *closing* a *opening* (obrázek 3.10).



Obrázek 3.10: Operace closing a opening

Mějme objekt, který není celistvý a obsahuje například otvory. Pokud jsou otvory dostatečně malé, lze pro zacelení objektu využít operaci Closing. Ta nejprve provede dilataci obrazu A_D. Tím získáme zvětšený objekt, který už neobsahuje otvory. A poté aplikuje erozi $(A_D)_E$, kterou vrátí objektu původní velikost.

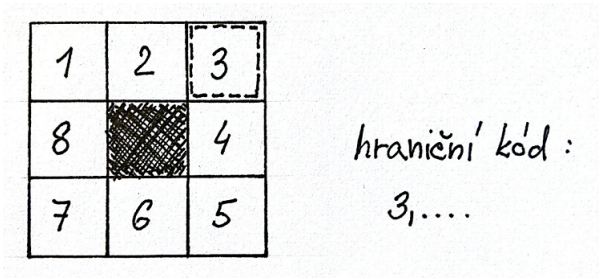
Opening se využívá, pokud je potřeba zjednodušit hranici objektu, nebo se v obraze vyskytují drobné objekty, které mají být odstraněny. Prvně se použije eroze A_E, která vrátí zmenšený objekt, ale s vyhlazenou hranicí a která odstraní malé objekty v pozadí. Jako druhý krok se provede dilatace $(A_E)_D$. Ta zase vrátí objekt do původní velikosti.

Morfologické operace nejsou citlivé na aditivní šum, což je jejich velkou výhodou.

3.5. Rozpoznávání objektů

Rozpoznávání objektů je hlavním cílem analýzy obrazu. Objekty lze chápat třemi různými způsoby. Můžeme analyzovat pouze hranici objektu, můžeme celý objekt chápat jako množinu bodů v rovině nebo ho můžeme chápat jako množinu pixelů.

K reprezentaci objektu jeho hranicí je třeba uložit souřadnice jednoho bodu hranice a k tomu uložit tzv. *hraniční kód*. Hraniční kód je posloupnost přechodů z pixelu na další pixel přes celou hranici objektu. Získání jednotlivých složek kódu je naznačeno na obrázku 3.11.



Obrázek 3.11: Hraniční kód

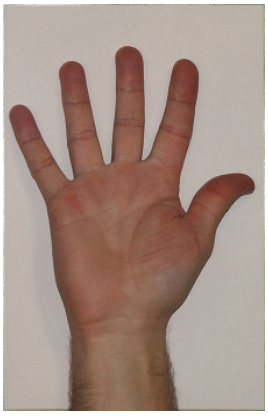
4. Praktická část

Cílem této práce je nalézt vhodnou metodu, která by z fotografie ruky dokázala určit její přibližný povrch. Vstupem je tedy fotografie a pokud z ní chceme získat rozměry potřebné pro výpočet povrchu, musíme na ni zaznamenat měřítko. Fotografie bude procházet algoritmem, který je popsán v následujících kapitolách a který předpokládá specifický vstupní obraz. Přesnost výpočtu bude tedy náchylná na velké změny polohy ruky v obraze a na kvalitu fotografie. Proto vstupní obraz musí splňovat tato kritéria:

- fotit na bílém podkladovém papíru s kalibračním rámečkem o rozměrech 180x280mm
- pravá ruka dlaní k fotoaparátu
- umístit ruku na střed
- holé předloktí, tj. bez rukávu apod., kvůli detekci zápěstí
- prsty roztažené od sebe
- prostředníček rovně s předloktím
- volba vhodného osvětlení
- fotit z dostatečné vzdálenosti k potlačení vlivu perspektivy

Poznámka 4.0.1. Při focení vzniká středové promítání, které nezachovává délky. Ale v případě dostatečně velké vzdálenosti fotoaparátu lze promítání považovat za rovnoběžné. Proto je mezi kritérii uveden i požadavek, aby fotografie byla pořízena alespoň ze vzdálenosti 2m. Jestliže výška/šířka ruky je cca 110mm a její tloušťka (tj. výška od podložky) je cca 20mm, pak chyba, které se při focení z 2m dopustíme vlivem středového promítání, je rovna 0,55mm. Tato chyba je dostatečně malá na to, abychom ji mohli zanedbat.

Po pořízení fotografie ji ořízneme dle rámečku na podkladovém papíru, který má rozměr 180x280mm. Formát digitální fotografie nastavíme na 900x1400 pixelů, takže jeden pixel se bude rovnat 0,2mm. Tato informace je důležitá a využijeme ji v posledním kroku výpočtu povrchu při jeho převodu na mm^2. Nakonec barevný obraz převedeme na obraz ve stupních šedi. Všechny výpočty a úpravy budou prováděny pomocí softwaru Matlab.



Obrázek 4.1: Příklad vstupního obrazu

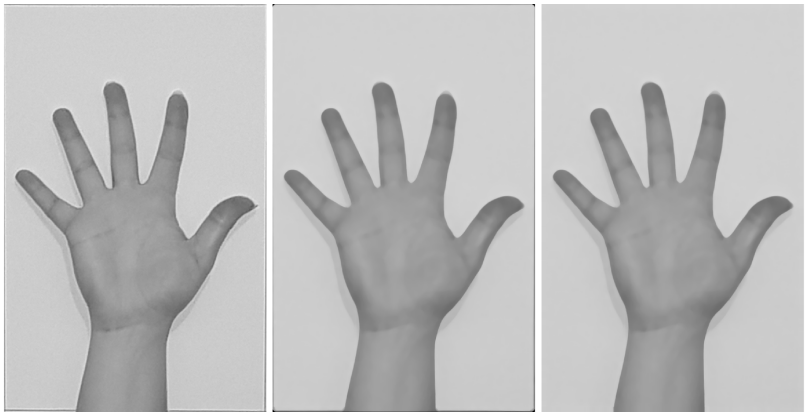
4.1. Určení jednopixelové hranice ruky

Jelikož je na vstupu fotografie, je v první řadě nutné zpracovat obraz tak, aby z něj bylo možné vyčíst důležité informace, tedy určité rozměry ruky. A k tomu je třeba nalézt jednopixelovu hranici objektu. Na této hranici lze určit tzv. body zlomu, tj. například špičky a úpatí prstů. Pokud známe všechny potřebné body, nahradíme části ruky co nejpřesnějšími geometrickými útvary a spočítáme jejich povrch. Konkrétní body a útvary jsou popsány v kapitole 4.2.

4.1.1. Segmentace obrazu - prahování

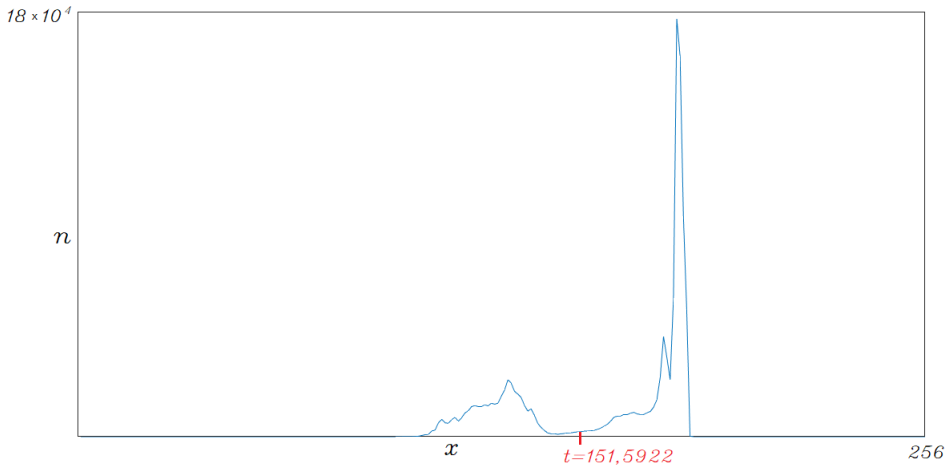
V obraze se nachází tmavý objekt na světlém pozadí, proto k jeho segmentaci využijeme metodu prahování. Ne každý obraz však bude ideální. Je tedy nutné si ho nejprve upravit tak, aby rozdílnosti kvality fotografií neměly vliv na výsledky segmentace.

Mediánový filtr je vhodný na filtraci šumu v obraze a zároveň zachovává hrany objektů. Na obraz tedy aplikujeme mediánový filtr o velikosti 21x21 pixelů. Dále víme, že na okrajích obrazu se nenachází žádná důležitá informace, takže pixely nacházející se v 15-ti pixelovém pásu po okrajích obrazu můžeme s klidným svědomím odstranit (přidělit jim jinou hodnotu). Jak ale zvolíme novou hodnotu pixelu? Přestože víme, že pozadí obrazu je světlé, nemůžeme zvolit libovolně světlou. Pak by se mohlo stát, že tyto nově vložené hodnoty pixelů zásadně posunou prahovou hodnotu při prahování obrazu a získáme pak neodpovídající objekt. Proto přepíšeme jejich hodnoty na hodnoty pixelů ležících hned vedle na řádku (pak ve sloupci) za tímto pásem. Tento krok je důležitý, protože se v okrajích můžou vyskytovat zbytky tmavého rozměrového rámečku, který jsme nepřesně ořízli a tyto tmavé pixely by mohly znovu nepříznivě ovlivnit prahování.

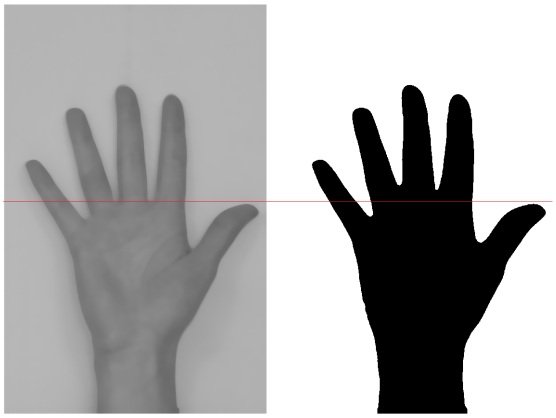


Obrázek 4.2: Vstupní obraz; s mediánovým filtrem; po odstranění okrajů

Histogram obrazu (viz 4.3) je bimodální, přesto nemůžeme využít samotné prahování. Pokud bychom použili jednu prahovou hodnotu na celý obraz, neošetřili bychom možné stíny, nebo naopak přesvětlený objekt v původní fotografii. Např. mezery mezi prsty by se mohly začít zkracovat, jak je vidět na obrázku 4.4.



Obrázek 4.3: Histogram obrazu a jeho prahová hodnota t



Obrázek 4.4: Obraz prahovaný jednou prahovou hodnotou

Využijeme tedy adaptivní prahování, tzn. pro každý pixel a pro jeho konkrétně zvolené okolí vytvoříme histogram, který by měl být znovu bimodální. Pak nejdeme práh a zařadíme pixel buď do objektu nebo do pozadí. Volba velikosti okolí je zásadní. Pokud je zrovna zvoleno okolí, v němž se nenachází žádný objekt a vybraný obraz je tedy téměř jednolitý, prahová hodnota bude zcestná. Pak by cyklus zakreslil objekty i tam, kde žádné nejsou, nebo naopak rozdělil objekt na více objektů.

Naším hlavním cílem je, aby obrys objektu zůstal co nejlépe zachovaný. Použijeme-li příliš velké okolí pro adaptivní prahovaní, výsledek bude podobný, jako bychom prahovali celý obraz najednou a prsty se začnou vlivem stínů spojovat (viz okolí 151 a 201 pixelů na obrázcích 4.5d a 4.5e). Naopak použijeme-li příliš malé okolí, hrozí, že se v blízkosti hranice najde objekt a pozadí podobných intenzit světla, histogram nebude výrazně bimodální, prahová hodnota bude vyšší, než by měla být a objektový pixel bude považován za pozadí. Pak by se hranice objektu mohla v některých místech roztrhnout (viz okolí 51 pixelů na obrázku 4.5b). Využijeme tedy kompromisu 101x101 pixelů velké okolí (viz obrázek 4.5c).

(a) obraz (b) okolí 51p (c) okolí 101p (d) okolí 151p (e) okolí 201p

Obrázek 4.5: Adaptivní prahování s použitým okolím o velikosti

Jelikož výsledek prahování není stále uspokojující, přidáme další úpravy, které se budou provádět uvnitř každého cyklu prahování, tj. pro každý pixel a jeho okolí. Můžeme si všimnout, že největší nepřesnosti vznikají kvůli stínům, protože jsou tmavé, podobně jako objekt. Jejich hodnota jasu bývá větší než prahová hodnota a mají tak tendenci se přiřazovat k objektu a zvětšovat ho. Proto jednou z úprav bude posunutí prahové hodnoty do tmavších hodnot o 5%. Přestože jsme na začátku zpracování už jednou filtrovali obraz mediánovým filtrem, který potlačil v obraze šum, použijeme mediánový filtr znovu uvnitř cyklu. Po aplikaci tohoto filtru o velikosti 5x5 pixelů bude vybraná část obrazu ještě víc rozostřena. Plochy pozadí a objektu budou jednolitější, ale hranice mezi nimi zůstane na stejném místě. V histogramu budou vyšší četnosti pixelů stejného jasu a tak z něj bude možné jednoznačněji určit práh mezi objektem a pozadím. Efekt těchto úprav je zobrazen na obrázcích 4.6.

(a) bez úprav (b) posunutý práh (c) med. filtr (d) obě úpravy

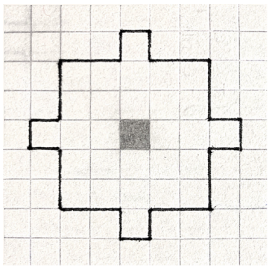
Obrázek 4.6: Efekty prahování s posunutým prahem o 5% a s použitým mediánovým filtrem o velikosti 5x5 pixelů

4.1.2. Nutné úpravy k výsledné hranici objektu

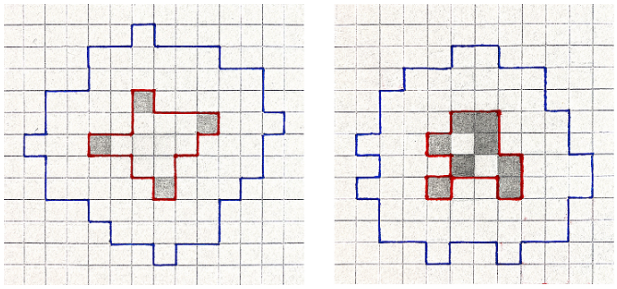
Přestože jsme metodu segmentace ladili tak, aby na výstupu byl pouze jeden objekt, který obsahuje celou hranici ruky, nemůžeme předpokládat, že tomu tak vždy bude. Pokud vstupní fotografie bude v opravdu špatné kvalitě, může se stát, že se objekt ruky rozdělí na více částí, nebo výsledný prahovaný obraz bude obsahovat i jiné objekty nalezené v pozadí ruky. Proto provedeme ještě několik úprav po samotné segmentaci.

Poznámka 4.1.1. Pro lepší demonstraci významu jednotlivých kroků úprav využijeme špatně vyprahovaných obrazů, které jsme získali při ladění segmentace. Zobrazují tak vše, co by se mohlo stát a co musíme ošetřit.

Pro začátek obrátíme binaritu obrazu tak, aby objekty měly hodnoty pixelů rovny jedné, tedy byly bílé. Dalším krokem bude operace closing (dilatace a následná eroze), která zacelí objekty. Jelikož jsou naše objektové pixely reprezentovány vysokou hodnotou, operace closing znamená aplikace filtru typu maximum a poté aplikace filtru typu minimum. Filtrovat budeme s „kruhovým“ okolím o poloměru 3 pixelů (viz obrázek 4.7). Příklady toho, jak velké nespojitosti v objektech closing s takovým okolím spojí, můžeme sledovat na obrázku 4.8, kde tmavě šedé čtverečky vyjadřují objektové pixely, modrá hranice znázorňuje hranici objektu po aplikaci max-filtru a červená pak znázorňuje hranici výsledného objektu po aplikaci min-filtru a tedy celé operace closing.



Obrázek 4.7: Kruhové okolí o poloměru 3 pixelů

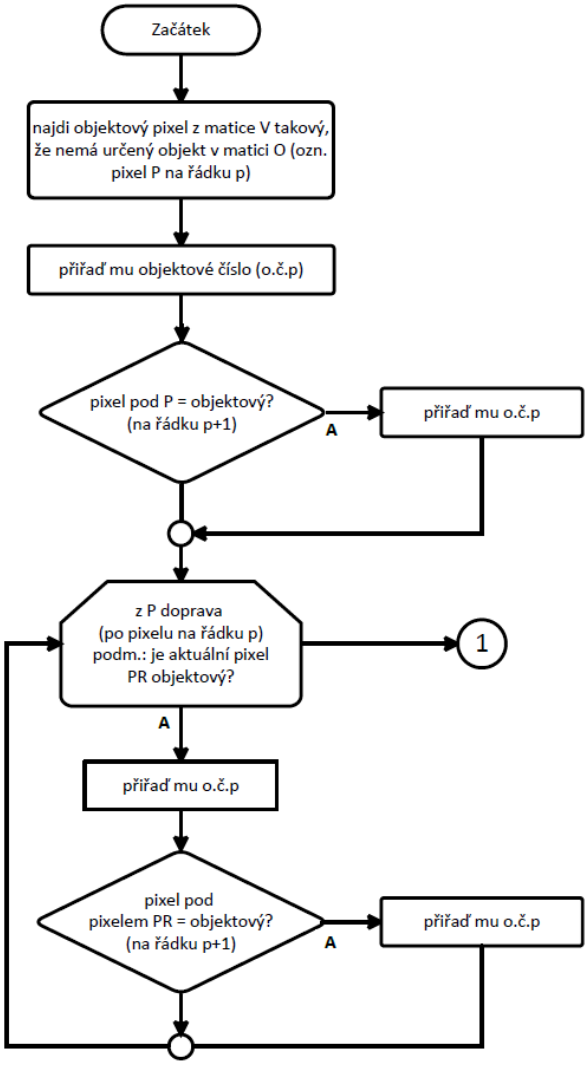


Obrázek 4.8: Příklady closingu s použitým kruhovým okolím o poloměru 3 pixelů

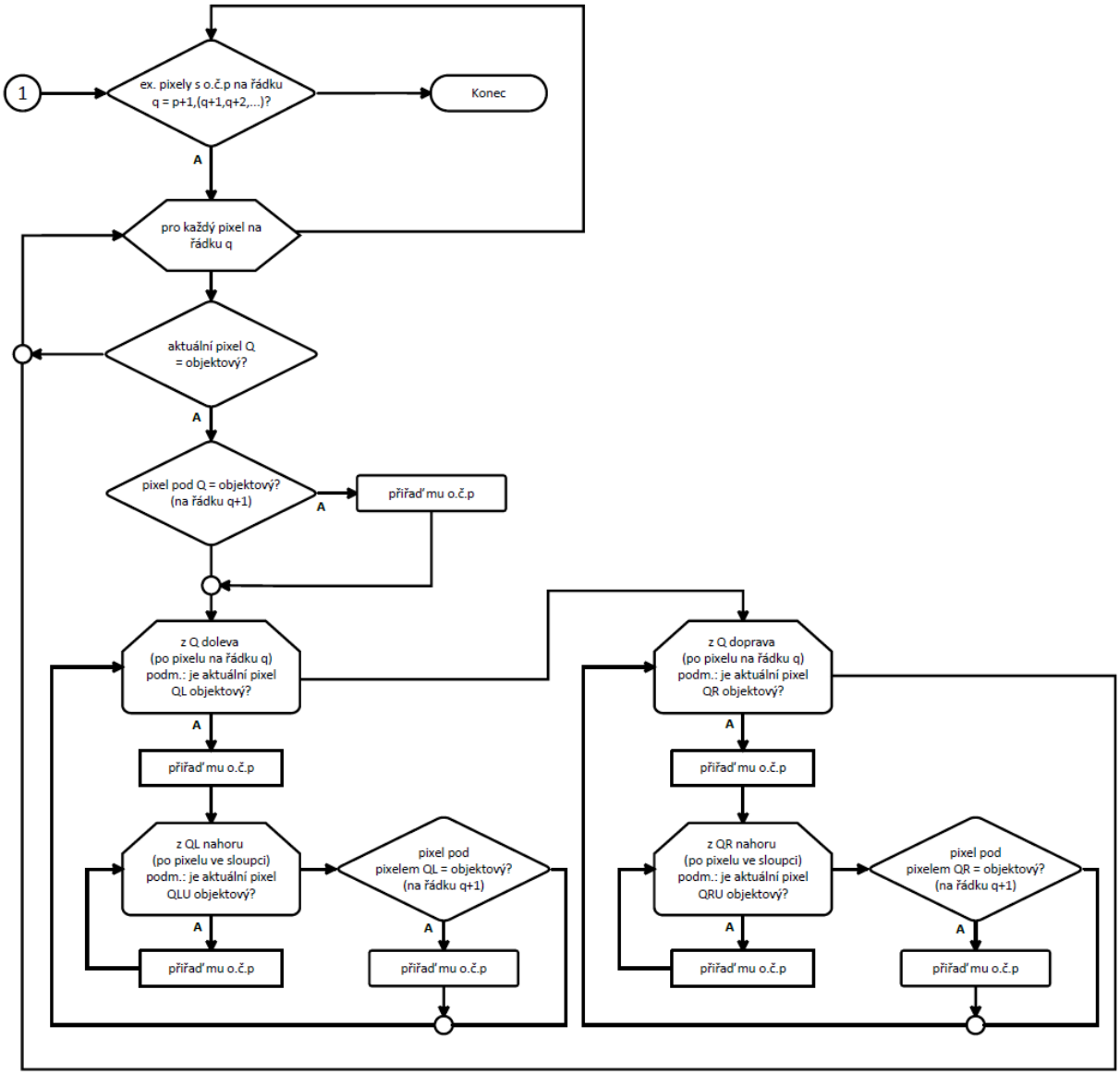
Při segmentaci se nám mohou v okrajích zobrazit objekty, které v obraze ve skutečnosti nejsou. Proto znovu odstraníme 15-ti pixelový pás podél krajů celého obrazu. Slovem odstraníme se zde myslí skutečnost, že přepíšeme hodnoty pixelů na nuly (hodnoty pixelů pozadí). Další malou úpravou bude odstranění samostatných pixelů. Tak nazveme pixely, který nemají v čtyřsousednosti jiný pixel, neboli nejsou s jiným pixelem 4-spojitelné.

Algoritmus pro vyhledávání objektů v obraze

Pomocí algoritmu popsaného na obrázcích 4.9 a 4.10 najdeme všechny objekty v obraze, které budou odlišeny tzv. objektovými čísly. Maticí V označíme vstupní obraz tvořený nulami a jedničkami, kde jedničky jsou pixely objektů. Výstupní matice O je stejně veliká jako matice V. Na pozicích objektových pixelů v matici V, bude matice O obsahovat objektová čísla pixelů o.č.p rovna 1 až n, kde n je celkový počet objektů v matici V. Na pozicích, kde pixely matice V jsou nulové budou i hodnoty matice O nulové. Názorné zobrazení matice O můžeme sledovat na obrázcích 4.11 a 4.12, kde objekty s různými o.č.p jsou od sebe odlišené barevně.



Obrázek 4.9: Vývojový diagram algoritmu vyhledávajícího objekty (část 1.)



Obrázek 4.10: Vývojový diagram algoritmu vyhledávajícího objekty (část 2.)

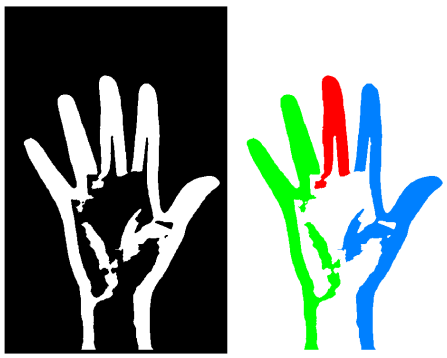
Třetí částí algoritmu je oprava rozdělených objektů. Jak je vidět na obrázku 4.11a, 1. a 2. část neošetří všechny směry při propojování pixelů jednoho objektu. Proto v posledním kroku algoritmu projdeme matici O a do nulového vektoru o délce počtu nalezených objektů zapíšeme změny v o.č.p. Nalezneme-li pixel, který má ve svém čtyřsousednosti pixel s jiným o.č.p, pak do změnového vektoru na pozici většího o.č.p zapíšeme hodnotu menšího o.č.p. Po tom co takto projdeme celou matici O, opravíme o.č.p dle změnového vektoru. Čteme ho odzadu a pro každou nenulovou hodnotu m na k-té pozici opravíme v matici O všechny o.č.p $= k$ na správné o.č.p $= m$. Příklad opravy objektů a tedy výsledné nalezení všech objektů v obraze tímto algoritmem je na obrázku 4.11b.

(a) Zobrazení nalezených objektů 1. a 2. částí algoritmu

(b) Zobrazení nalezených objektů celým algoritmem

Obrázek 4.11: Nalezení všech objektů v obraze

Po nalezení všech objektů v obraze vidíme, že kromě velkých částí ruky se v pozadí nachází plno malých nevýznamných objektů, které v obraze nemají být. Experimentálně jsme stanovili, že objekty, který se skládají z méně než 20000 pixelů jsou nevýznamné a můžeme je odstranit. Výsledné hledání objektů je vidět na obrázku 4.12.



Obrázek 4.12: Výsledné barevné zobrazení nalezených objektů ruky

Získání uceleného objektu

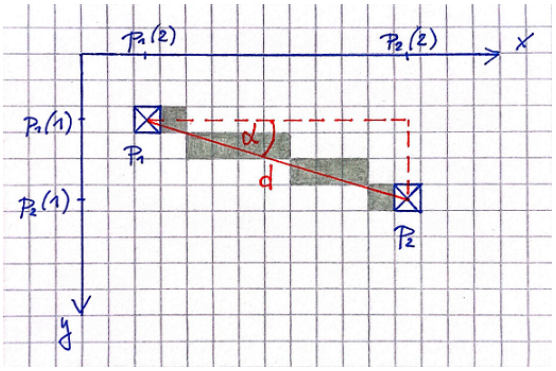
Cílem této kapitoly je nalézt hranici ruky. Jak ale stanovit výslednou hranici, když zatím nemáme ani objekt ruky jako celek? Nejprve musíme zjistit jak správně propojit nalezené části, jak určit jejich pořadí a v neposlední řadě jak určit, které části hranic jednotlivých objektů patří do celkové hranice ruky. Následným postupem obraz upravíme tak, aby jsme v každém případě byli schopni hranici nalézt.

Připomeňme si, že matice V je výsledný binární obraz obsahující objekty ruky a matice O je matice, která odlišuje objekty v matici V pomocí o.č.p.

1. Vytvoříme hraniční matici H, která bude obsahovat 8-spojitelnou hranici všech objektů z matice V tak, že všechny pixely, které ve svém okolí $\mathcal{N}_4(r,s)$ mají maximálně tři pixely, jsou hraniční.

2. Vektor hranic $vektorH$ bude mít 3 složky. První dvě budou obsahovat pozici v matici $[r,s]$ a třetí složka bude o.č.p daného pixelu. Tento vektor vytvoříme pomocí matic H a O, kde $H[r,s] = 1$ a $O[r,s] =$ o.č.p.

3. Matici obálky VO vytvoříme propojením všech pixelů v matici H. Pro každé dva pixely p_1 a p_2 vypočítáme vzájemnou vzdálenost d a úhel α. Pak každému pixelu, který protne spojnice pixelů p_1 a p_2, udělíme v matici VO hodnotu jedna. Viz příklad na obrázku 4.13, kde pixely p_1, p_2 a všechny šedé budou v matici VO označeny hodnotou jedna.



Obrázek 4.13: Příklad propojení dvou pixelů

Vytvoření spojnic a tedy výpočet vzdálenosti d a úhlu α provedeme následujícím způsobem.

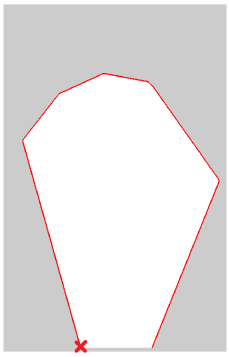
$$\begin{aligned} d &= \sqrt{(p_2(1) - p_1(1))^2 + (p_2(2) - p_1(2))^2} \\ \alpha &= \arctan\left(\frac{p_2(1) - p_1(1)}{p_2(2) - p_1(2)}\right) \\ line &= 0 : 1 : d \\ y &= p_1(1) + line \cdot \sin\alpha \\ x &= p_1(2) + line \cdot \cos\alpha \end{aligned} \tag{4.1}$$

Pro výpočet úhlu však nepoužijeme klasickou funkci Arkus tangens, nýbrž využijeme funkce *atan2* (tj. Čtyř-kvadrantový arkus tangens), kterou software Matlab nabízí. Tato funkce vrací hodnoty v uzavřeném intervalu $\langle -\pi, \pi \rangle$. Vyřešíme tak všechny možnosti úhlů při vzájemných pozicích pixelů p_1 a p_2.

Obrázek 4.14: Matice V a matice její obálky VO

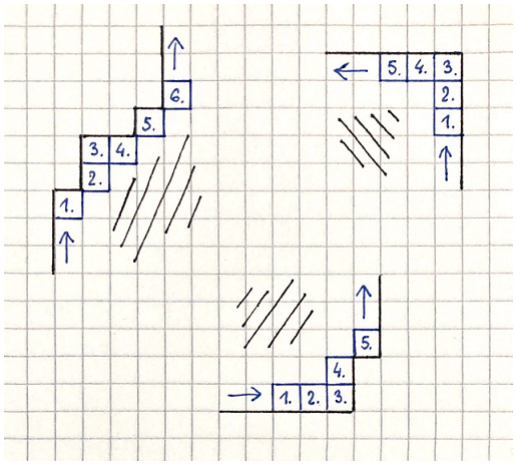
4. Vytvoříme hraniční matici obálky VO a označíme ji HO. Bude obsahovat její 8--spojitelnou hranici, tedy z matice VO vybereme pixely, které ve svém okolí $\mathcal{N}_4(r, s)$ mají maximálně tři pixely.

5. Uspořádaný vektor hranice obálky *vektorHO* bude obsahovat pozice hraničních pixelů v matici *HO*. Do tohoto vektoru však nezapíšeme pixely tvořící spodní „základnu“, což lépe vyjadřuje obrázek 4.15, kde červená čára vyznačuje hranici zapsanou do vektoru.



Obrázek 4.15: Uspořádaný vektor hranice obálky *vektorHO*

Na začátku úprav jsme odstraňovali 15-ti pixelový pás po okrajích celého obrazu, proto víme, že spodní okraj obálky leží na 16. pixelu odspodu. Na tomto řádku $[r, :]$ najdeme první hraniční pixel matice *HO* (v obrázku 4.15 označený křížkem). Následně nalezneme druhý pixel hranice tak, že neleží na $[r, :]$ řádku, ale na řádku $[r - 1, :]$ a zároveň je s prvním nalezeným pixelem 8-spojitelný. Abychom nalezli třetí pixel hranice, zkontrolujeme všechny pixely v osmisousednosti druhého. Nalezneme zde dva (případně tři) pixely z nichž třetí hraniční pixel bude ten, který se ve vektoru *vektorHO* ještě nenachází. Dále pokračujeme algoritmem, který k naposled zapsanému pixelu hledá 8-spojitelné pixely v matici *HO*. Vždy najde dva (případně tři) a zapíše ten, který se do vektoru nezapsal v předchozích dvou krocích, tj. algoritmus ošetří, abychom se nevraceli zpět. Poslední pixel vektoru *vektorHO* je takový pixel, který znovu leží na řádku $[r, :]$ (nedostaneme se zpět do počátečního pixelu, pouze na stejný řádek).
Důvodem, proč v algoritmu porovnáváme pixel s pixely z dvou předchozích kroků jsou případy, kdy hranice obálky vypadá podobně jako na obrázku 4.16.



Obrázek 4.16: Příklady hranice obálky

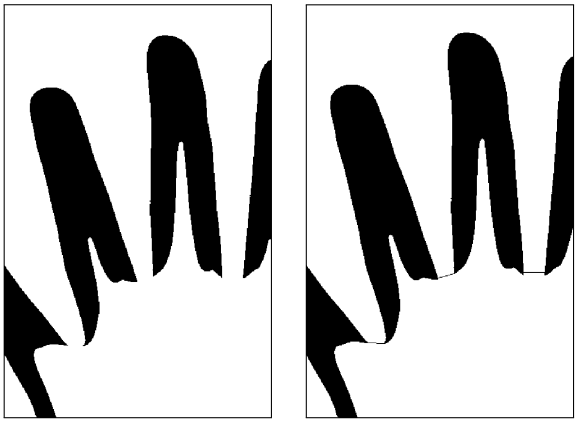
Algoritmus, který by porovnával pixel pouze s jedním předchozím pixelem, by se v 2., 3. a 4. pixelu (viz obrázek 4.16) zacyklil. Zajistíme-li navíc, aby hledal předně ve čtyřsousednosti pixelu, najde pak vždy správné řazení hranice.

6. Nyní můžeme vybrat objekty, které opravdu tvoří hranici ruky a můžeme určit pořadí, ve kterém na sebe mají navazovat. Budeme procházet vektor hranice obálky *vektorHO* a k tomu budeme potřebovat hlavní matici V a objektovou matici O. Pokud narazíme ve vektoru *vektorHO* na pixel, který je v matici V označen jako objektový pixel, zapíšeme jeho o.č.p (najdeme v matici O) do vektoru pořadí. Narazíme-li na další takový pixel se stejným o.č.p jako měl ten předchozí, přeskočíme ho a pokračujeme v procházení vektoru *vektorHO* dál. Narazíme-li na objektový pixel ve V s jiným o.č.p v O než měl pixel předchozí, zapíšeme ho na další pozici vektoru pořadí. Takto pokračujeme dokud nedojdeme na konec vektoru *vektorHO*.

7. Díky tomu, že už známe správné pořadí jednotlivých objektů, víme, který objekt s kterým potřebujeme propojit. Tedy můžeme najít jejich minimální vzdálenost a následně je v tomto místě propojit. Ve vektoru *vektorH* máme uložené pozice hraničních pixelů všech objektů a k nim i jejich o.č.p. Pro dva objekty, které na sebe mají dle vektoru pořadí navazovat, spočítáme vzdálenosti každý s každým pixelem a určíme tu minimální.

8. Než začneme propojovat jednotlivé objekty, spojíme spodní část ruky (zápěstí), aby objekt ruky byl uzavřený. Jednoduše spojíme první a poslední pixel ležící na řádku nad spodním 15-ti pixelovém pásu matice V.

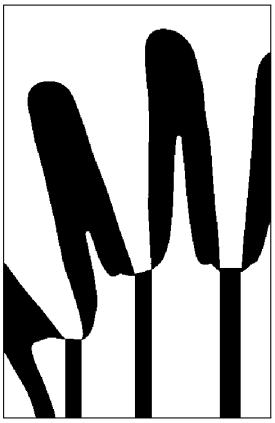
9. Z předchozích kroků známe pro každé dva objekty dva konkrétní pixely p_1 a p_2, které se mají propojit a známe jejich vzdálenost d. Stačí tedy spočítat úhel α a stejně jako ve 3. kroku vytvořit spojnici pixelů p_1 a p_2, a tedy i spojnici objektů.



Obrázek 4.17: Propojení objektů (negativ)

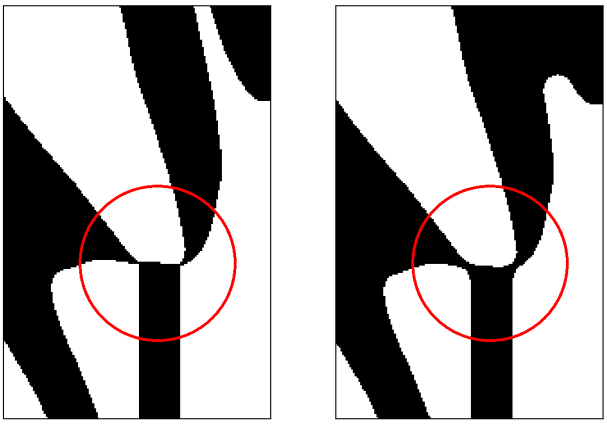
10. Než konečně definujeme hranici ruky, musíme provést ještě dvě úpravy, které se mohou zdát zbytečné. Dále si však popíšeme způsob určení hranice, který tyto kroky dovysvětlí.
    Pokud necháme spojnice mezi objekty jednopixelové, algoritmus hledání hranice bude mít v tomto místě víc možných směrů, kam dál postupovat a nenajde tak správnou hranici. Naštěstí se všechny nespojitosti objektů nacházejí mezi prsty a proto stačí aplikovat tyto kroky:

(a) Pod každý pixel, který vytvoříme spojením pixelů p_1 a p_2, budeme pixelům s hodnotou nula vkládat hodnotu jedna dokud nenarazíme na jiný objektový pixel (roven jedné). Také pod krajní pixely p_1 a p_2 vytvoříme takový sloupec hodnot.



Obrázek 4.18: Propojení objektů - rozšíření spojnice (negativ)

(b) Provedeme operaci closing (dilatace a eroze) s kruhovým okolím o poloměru 8 pixelů. Tímto se spojnice objektů v místě napojení rozšíří. Viz obrázek 4.19b.

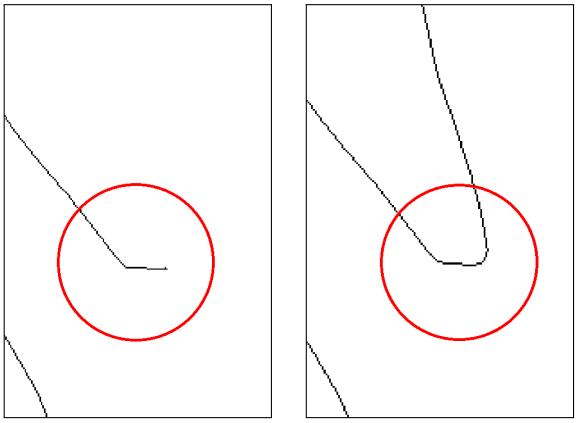


(a) bez closingu (b) s closingem

Obrázek 4.19: Propojení objektů - operace closing (negativ)

Příklad, jak tyto úpravy můžou ovlivnit výslednou hranici ruky ukazuje obrázek 4.20.

11. Poslední úpravou, kterou můžeme zlepšit autentičnost hranice ruky je odstranění vyčnívajících pixelů. Jsou to objektové pixely, které ve svém $\mathcal{N}_4(r, s)$ okolí mají pouze jeden další objektový pixel, nebo naopak pixely pozadí, které mají v čtyřsousednosti hned tři objektové pixely.

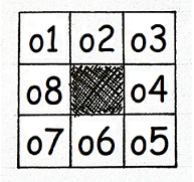


Obrázek 4.20: Příklad hranice bez úprav v 10. kroku a s předchozími úpravami (negativ)

Algoritmus pro nalezení hranice objektu

Nyní máme v matici V připravený objekt, jehož vnější hranice odpovídá hranici ruky. Následujícím algoritmem získáme matici HV obsahující 8-spojitelnou hranici ruky a vektor $vektorHV$, který bude obsahovat uspořádanou posloupnost pixelů hranice v matici HV.

Označme okolí pixelu p dle obrázku 4.21.



Obrázek 4.21: Označení okolních pixelů pixelu p

Jestliže pixel p je v n-tém kroku posledním pixelem zapsaným do matice HV a do vektoru $vektorHV$, pak jeho okolní pixely $o1$ až $o8$ budou rovny

$$\begin{aligned}
o1 &= [vektorHV(n,1) - 1, vektorHV(n,2) - 1],\\
o2 &= [vektorHV(n,1) - 1, vektorHV(n,2)\ \ \ \ \ ],\\
o3 &= [vektorHV(n,1) - 1, vektorHV(n,2) + 1],\\
o4 &= [vektorHV(n,1)\ \ \ \ \ , vektorHV(n,2) + 1],\\
o5 &= [vektorHV(n,1) + 1, vektorHV(n,2) + 1],\\
o6 &= [vektorHV(n,1) + 1, vektorHV(n,2)\ \ \ \ \ ],\\
o7 &= [vektorHV(n,1) + 1, vektorHV(n,2) - 1],\\
o8 &= [vektorHV(n,1)\ \ \ \ \ , vektorHV(n,2) - 1].
\end{aligned}$$

Nejprve najdeme první tři pixely hranice, což provedeme trochu jiným způsobem, než později všechny ostatní. U prvního pixelu postupujeme stejným způsobem jako u uspořádaného vektoru hranice obálky $vektorHO$ v 5. kroku. V matici V procházíme 16. řádek odspodu a první objektový pixel zleva, který najdeme, je naším prvním pixelem hranice. Druhý a třetí pixel hledáme také zvlášť, protože za prvé potřebujeme určit správný počáteční směr hranice a za druhé algoritmus hledá další pixel hranice v závislosti na třech posledních. Použijeme však jeho upravenou verzi, která hledá další pixel pouze v

závislosti na posledním pixelu. U druhého pixelu hranice vybíráme z okolních pixelů $o1$, $o2$ a $o3$ prvního pixelu a u třetího pouze z okolních pixelů $o1$, $o2$, $o3$, $o4$ a $o8$ druhého pixelu.

Algoritmus, který z matice V postupně vybírá hraniční pixely ve správném pořadí, se řídí maticí OK, která se v každém kroku n mění. Tato matice obsahuje pozice všech osmi okolních pixelů posledního zapsaného pixelu p do matice HV a do vektoru $vektorHV$. Další dva řádky matice jsou počty objektových pixelů v okolí $\mathcal{N}_4(r,s)$ a v okolí $\mathcal{N}_8(r,s)$ daného okolního pixelu o v matici V. Např. první sloupec matice OK obsahuje svislou pozici pixelu $o1$, vodorovnou pozici pixelu $o1$, počet objektových pixelů v jeho okolí $\mathcal{N}_4^{o1}(o1(1),o1(2))$ a počet objektových pixelů v jeho okolí $\mathcal{N}_8^{o1}(o1(1),o1(2))$.

Počáteční tvar matice OK v n-tém kroku je

$$OK = \begin{pmatrix} o1(1) & o2(1) & o3(1) & o4(1) & o5(1) & o6(1) & o7(1) & o8(1) \\ o1(2) & o2(2) & o3(2) & o4(2) & o5(2) & o6(2) & o7(2) & o8(2) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{4.2}$$

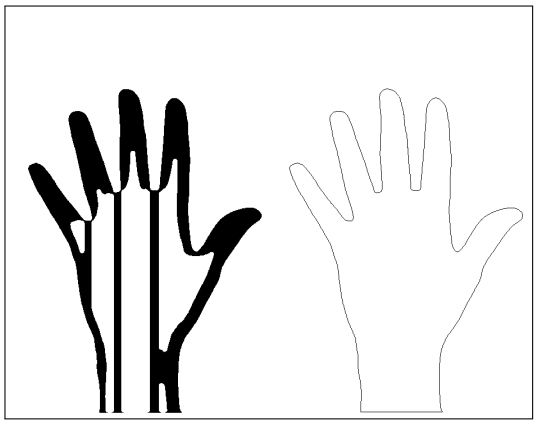
Pro každý okolní pixel o posledního pixelu p zjistíme, zda je to objektový pixel a zda se nejedná o hraniční pixel z $n-1$ kroku nebo z $n-2$ kroku. Jestliže tyto podmínky splňuje, spočítáme jeho okolní objektové pixely v $\mathcal{N}_4^o(o(1),o(2))$ a v $\mathcal{N}_8^o(o(1),o(2))$. Hodnoty 100 ve třetím a čtvrtém řádku značí, že pixel o jednu z těchto podmínek nesplňuje a určitě se tedy nejedená o další pixel hranice.

```
for o=1:8
    if (
      %je pixel objektovy
      V(OK(1,o),OK(2,o))==1
      &&
      %je ruzny od pixelu - 1 krok zpet
      (vektorHV(n-1,1)~=OK(1,o) || vektorHV(n-1,2)~=OK(2,o))
      &&
      %je ruzny od pixelu - 2 krok zpet
      (vektorHV(n-2,1)~=OK(1,o) || vektorHV(n-2,2)~=OK(2,o))
      )
            OK(3,o)= V(OK(1,o)-1,OK(2,o))
                    +V(OK(1,o)+1,OK(2,o))
                    +V(OK(1,o),OK(2,o)-1)
                    +V(OK(1,o),OK(2,o)+1);
            OK(4,o)=OK(3,o)
                    +V(OK(1,o)-1,OK(2,o)-1)
                    +V(OK(1,o)-1,OK(2,o)+1)
                    +V(OK(1,o)+1,OK(2,o)-1)
                    +V(OK(1,o)+1,OK(2,o)+1);
    else
            OK(3,o)=100;
            OK(4,o)=100;
    end
end
```

Druhou částí algoritmu vyhodnotíme získané informace o okolních potenciálních pixelech hranice. Prvně, pixel o musí být hraniční tzn. v jeho okolí $\mathcal{N}_4^o(o(1), o(2))$ musí být maximálně tři další objektové pixely. Touto podmínkou vyřadíme rovnou všechny pixely, které se již v hranici nacházejí nebo nejsou objektové, protože mají $OK(3, o) = 100$. Druhá podmínka se v algoritmu nachází kvůli speciálním případům tvaru hranice. Jedná se o případy popsané na obrázku 4.16, kde pixel p nemá ve svém okolí pouze dva hraniční pixely, ale rovnou tři. Tato podmínka vybere správný pixel tak, aby se algoritmus nezacyklil.

```
for o=1:8
    if (
      %je pixel hranicni
      OK(3,o)<=3
      &&
      %je v jeho osmisousednosti nejmene pixelu
      OK(4,o)==min(OK(4,:))
      )
            vektorHV(n+1,:)=[OK(1,o),OK(2,o)];
            HV(OK(1,o),OK(2,o))=1;
            break;
      end
end
```

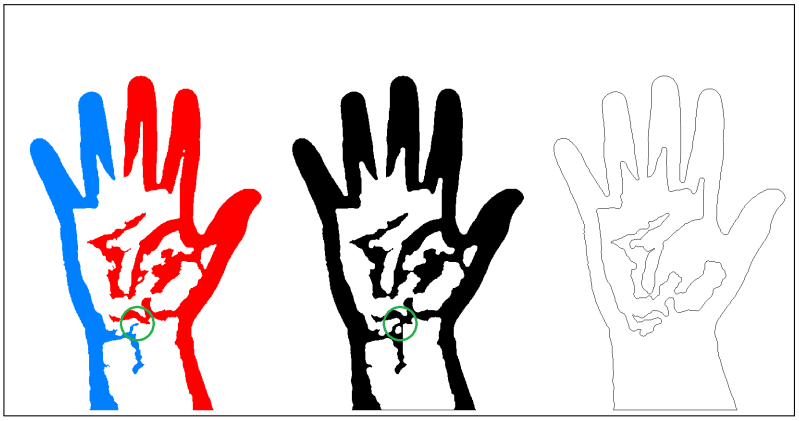
Algoritmus se ukončí, pokud je nově nalezený pixel roven poslednímu pixelu vnější hranice, který leží na stejném řádku matice V jako první pixel hranice. V závěru stačí tyto dva pixely spojit, abychom získali uzavřený objekt ruky.



Obrázek 4.22: Zobrazení objektu v matici V a jeho hranice v matici HV (negativ)

Opravy hranice objektu

Co když propojení objektů v 9. kroku neproběhne správně? Co když se objekty propojí na jiném místě než očekáváme, protože existují jiné pixely s minimální vzdáleností. Pak by výsledná hranice procházela vnitřkem ruky a neodpovídala skutečnosti (viz obrázek 4.23). Proto nakonec hranici objektu ještě opravíme následujícím postupem.



Obrázek 4.23: Příklad objektů, kterým zvolený postup nalezení hranice nestačí (negativ)

Z vektoru hranice $vektorHV$ vytvoříme matici M, která bude obsahovat vzdálenosti každého pixelu p_1 s každým pixelem p_2 a bude striktně dolní trojúhelníková. Cílem je nalézt takové pixely p_1 a p_2, jejichž vzdálenost d v obraze V je malá a zároveň se ve vektoru $vektorHV$ nacházejí daleko, tj. jejichž indexová vzdálenost hranice je velká. Pokud se propojením těchto pixelů dostatečně zkrátí hranice objektu, pak se nám podařilo uzavřít objekt na správném místě, a to mezi prsty.

Poznámka 4.1.2. Všechny následující hodnoty byly určeny experimentálně.

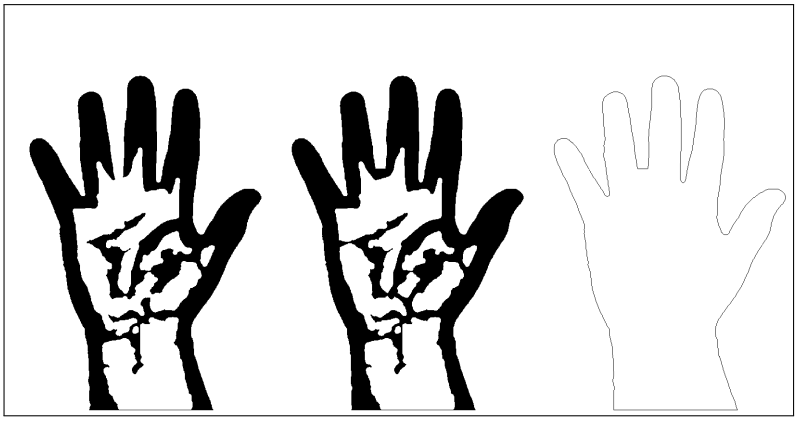
Počáteční vzdálenost vzd, která je maximální hranicí vzdáleností d pixelů p_1 a p_2 z matice M je $vzd = 15$. Tu v každém kroku zvětšíme, dokud se délka původní hranice $delkaHVSpred$ nezkrátí alespoň o 40%, nebo dokud vzdálenost $vzd \leq 35$. V každém kroku tedy vybereme z matice všechny dvojice pixelů jejichž $d \leq vzd$ a pokud je jejich indexová vzdálenost hranice větší než $20 \cdot vzd$, pak vytvoříme jejich spojnice stejně jako v 3. kroku (viz. obrázek 4.13 a rovnice 4.1). Dále použijeme Algoritmus pro nalezení hranice objektu a zjistíme tak novou délku hranice $delkaHVS$.

```
1  delkaHVS=delkaHV;
2  delkaHVSpred=delkaHVS;
3  vzd=15;
4
5  %pokracuj dokud vzd neni rovno 35
6  while vzd<=35
7      %pokracuj dokud se hranice nezkrati alespon o 40%
8      if delkaHVS<0.6*delkaHVSpred
9          break;
10     end
11
12     delkaHVSpred=delkaHVS;
13     for p=1:delkaHV
14         for q=1:p
15             %je vzdalenost p1 a p2 mensi nebo rovna vzd
16             if M(p,q)<=vzd
17                 %je index. vzdalenost vetsi nebo rovna 20*vzd
18                 if abs(p-q)>=20*vzd
19                     p1=[vektorHV(p,1) vektorHV(p,2)];
```

```
20                      p2=[vektorHV(q,1) vektorHV(q,2)];
21                      #SPOJENI PIXELU P1 A P2
22                  end
23              end
24          end
25      end
26
27      vektorHVS=zeros(delkaHVS,2);
28      HVS=zeros(vel(1),vel(2));
29      #ALGORITMUS PRO NALEZENI HRANICE OBJEKTU
30
31      delkaHVS=sum(sum(HVS));
32      vzd=vzd+1;
33  end
```

Ve většině případů získáme správnou hranici ještě před provedením této opravy. Avšak nemusíme se bát, že i správně nalezenou hranici by tento krok ovlivnil a začal by např. zaplňovat mezery mezi prsty. Díky podmínce vzdálených indexů ve vektoru hranice se žádné pixely nepropojí. Zároveň tomu brání i vzdálenost pixelů v obraze, která je ohraničena maximální hodnotou $vzd = 35$.

Jestliže hranice objektu před touto úpravou byla špatně nalezená, jako je např. hranice na obrázku 4.23, pak výsledek tohoto kroku bude objekt s hranicí uzavřenou a podobající se co možná nejlépe originálu. Výsledek úpravy můžeme sledovat na obrázku 4.24.



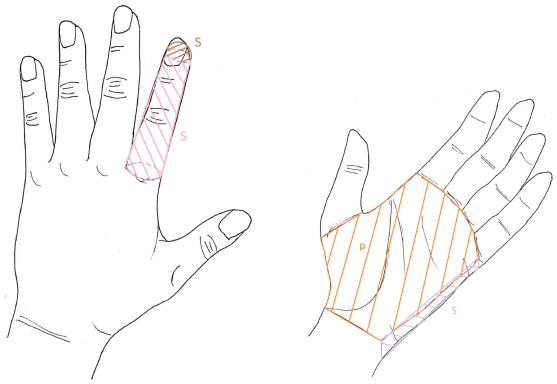
Obrázek 4.24: Oprava hranice špatně propojených objektů (negativ)

Jak již bylo řečeno na začátku kapitoly, většina částí a úprav ošetřuje případy špatně vyprahovaných obrazů a všechny obrázky v ní jsou testovací. Ukazují, že program dokáže zpracovat a najít hranici i v případě, že fotografie je ve špatné kvalitě a její prahovaný obraz neobsahuje pouze jeden celistvý objekt.

4.2. Definice parametrů ruky

Než začneme hranici ruky jakkoliv zpracovávat, musíme nejprve stanovit způsob, kterým se povrch ruky bude počítat. Nejintuitivnější způsob je pokusit se rozdělit ruku na menší části a připodobnit je konkrétním geometrickým útvarům a tělesům, u kterých je nám

znám vzorec pro výpočet obsahu či povrchu. Prsty mohou připomínat komolý kužel, který má na konci půl kouli. Dlaň má sice nepravidelný tvar, ale její obsah, jakožto obsah rovinného útvaru, dokážeme spočítat jednoduše. Tuto plochu přidáme dvakrát a přičtením boků dlaní získáme přibližný povrch 3D tělesa dlaně. Následující obrázek popisuje tento návrh výpočtu povrchu ruky, který vychází z předchozí bakalářské práce, na kterou tato práce navazuje. Jedná se o základ našeho výpočtu, který bude v dalším textu podrobněji popsán a upraven.



Obrázek 4.25: Původní návrh výpočtu povrchu ruky

V našem případě máme výhodu, že z 2D pohledu ruky známe každý pixel, z kterého se objekt skládá. K aproximaci tak neznáme pouze celkové rozměry těchto útvarů, ale známe všechny rozměry. Například průměr prstu v každé jeho pixelové výšce. Můžeme tedy daleko přesněji a po menších částech aproximovat a získat tak přesnější povrch ruky.

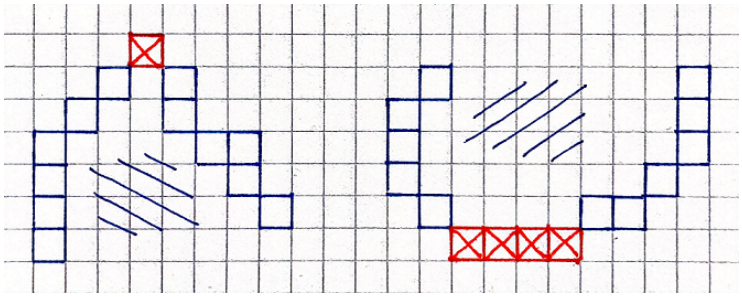
Potřebujeme tedy získat oddělené objekty prstů a objekt dlaně, což je cíl této kapitoly.

4.2.1. Body zlomu hranice ruky

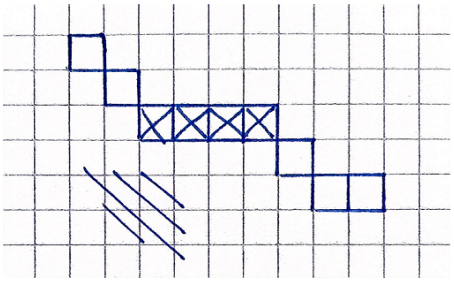
Bod zlomu hranice nazveme takový bod ležící na hranici, kde křivost křivky je v jeho okolí maximální. Jedná se tedy o špičky prstů, nebo naopak o jejich úpatí.

Jako první vytvoříme ze vstupní hranice objekt, abychom měli všechny základní matice objektu. Budeme procházet každý řádek matice hranice HVS a do matice VYS ukládat objektové pixely jak hranice, tak vnitřku objektu. Projdeme-li poprvé hranicí, pak se nacházíme uvnitř objektu. Projdeme-li znovu hranicí, pak jsme zpět v pozadí. Stačí tedy použít „přepínač“ (proměnnou, měnící hodnoty -1 a 1), který při průchodu hranicí změní znaménko. Podle něj pak určujeme, zda jsme v objektu či pozadí a zda mají pixely v matici VYS dostat hodnotu jedna či nula.

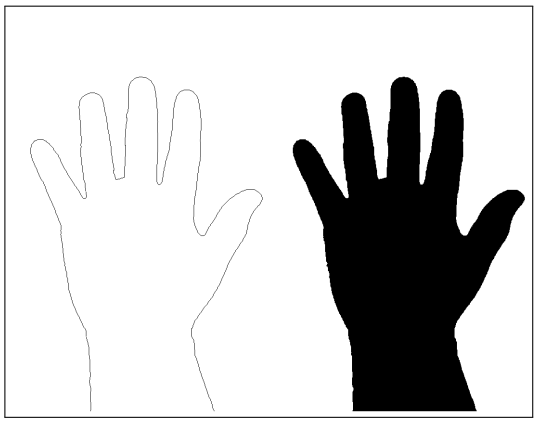
Výjimkou jsou pixely, nebo více pixelů za sebou, které se nacházejí na špičkách nebo v úpatí prstů. Projdeme-li tímto pixelem, nedostaneme se na opačnou stranu hranice. Příklady takových pixelů jsou na obrázku 4.26. Tyto pixely najdeme tedy přednostně a při vytváření matice VYS je nebudeme považovat za hranici. V těchto případech hledáme v řádku $[r, :]$ matice HVS jeden nebo více pixelů za sebou takových, že pixel před a pixel po jsou oba na řádku $[r-1, :]$, nebo na řádku $[r+1, :]$.



Obrázek 4.26: Příklad pixelů hranice, které nepovažujeme za hranici a přepínač se jimi nezmění



Obrázek 4.27: Příklad pixelů hranice, které přepínač mění



Obrázek 4.28: Matice HVS po odstranění spodní základny a matice VYS (negativ)

Křivost hranice

Budeme po částech aproximovat hranici ruky polynomem 4. stupně a počítat křivost v každém pixelu hranice. Následně tuto křivost vykreslíme do grafu a potom můžeme posoudit místa s největší křivostí.

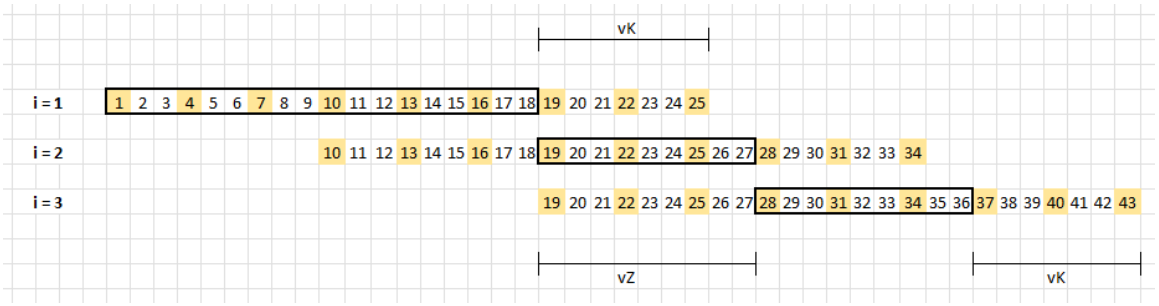
Vycházíme z vektoru $vektorHVS$, který obsahuje uspořádané pozice pixelů celé hranice. Odebereme z jeho konce pixely, které obsahují základnu ruky (viz matice HVS na obrázku 4.28). Jedná se o část hranice, která má pouze za úkol objekt uzavřít a nezajímá nás její křivost.

Zavedeme tři proměnné (proměnné mají pouze ilustrativní hodnoty k obrázku 4.29 a popisu výpočtu):

- $krok = 3$ znamená, že vybereme každý 3. pixel z posloupnosti pixelů hranice (začneme od 1. pixelu)

- $delkaKr = 9$ je počet pixelů z $krok$, které budeme aproximovat křivkou v jednom kroku algoritmu
- $pocetB = 3$ je počet pixelů z vybraných pixelů v $delkaKr$, ve kterých spočítáme křivost křivky (jelikož chceme znát křivost v každém pixelu celé hranice objektu, spočítáme ji i v jejich okolních pixelech nezahrnutých do aproximace)

Způsob výběru pixelů pro aproximování polynomem a výpočet křivosti popisuje následující obrázek, kde i je krok algoritmu a žlutě zvýrazněné pixely jsou vybrané pixely z vektoru $vektorHVS$ dle proměnné $krok$. V každém i-tém kroku algoritmu vybereme $delkaKr$ konkrétních žlutých pixelů, který aproximujeme křivkou. Křivost pak spočítáme pro $pocetB$ pixelů a pro jejich okolní pixely v černém tučném rámečku. Nepočítáme ji pro krajní pixely (viz čísti označené zkratkami vZ a vK), protože chceme získat křivost spojité hranice. Proto se také vybírané posloupnosti překrývají. Výjimkou je první a poslední krok algoritmu, ve kterém spočítáme i křivosti krajních pixelů vZ nebo vK, abychom získali křivost v každém pixelu hranice.

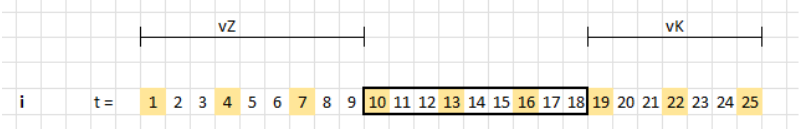


Obrázek 4.29: Výběr pixelů k aproximaci polynomem a výpočet jeho křivosti

Mluvíme zde o obrazu, o jeho matici a o pozicích hraničních pixelů v obraze. Na chvíli však z popisu orientace v matici, kde nejprve píšeme svislou a potom vodorovnou pozici pixelu, přejdeme k běžnému souřadnicovému systému bodů se souřadnicemi x a y. x-ová souřadnice bodu bude rovna druhé pozici pixelu a y-ová bude rovna první pozici pixelu.

Poznámka 4.2.1. Souřadnicový systém vůbec není nutné měnit, jelikož počítáme křivost, která je hodnotou nezávislou na souřadnicích a stejnou křivost má pixel v pozicích matice i bod v běžném souřadnicovém systému. Jde pouze o jasnější výpočty polynomu, jeho derivací a křivosti.

Mějme tedy posloupnost všech bodů hranice. Z této posloupnosti vybereme každý 3. bod ($krok$). Pak v každém i-tém kroku algoritmu pro výpočet křivostí hranice vybereme z těchto bodů 9 konkrétních bodů ($delkaKr$). Pro každý z nich známe souřadnice x a y a můžeme je tak aproximovat křivkou. Křivost pak nejlépe spočítáme dle vzorce 1.2, proto budeme křivku hledat v parametrickém vyjádření $f(t) = (x(t), y(t))$. Parametr t, bude pro nalezení aproximační křivky odpovídat pořadí bodu ve vybrané posloupnosti, tedy bude roven $t = [1, 4, 7, 10, 13, 16, 19, 22, 25]$.



Obrázek 4.30: Parametr t v i-tém kroku algoritmu

Chceme aproximovat 9 bodů polynomem 4. stupně, tzn. chceme body proložit křivkou tak, aby co nejlépe odpovídala jejím souřadnicím, což provedeme metodou nejmenších čtverců. V Matlabu pro výpočet polynomu n-tého stupně metodou nejmenších čtverců existuje funkce *polyfit(x,y,n)*. Pomocí této funkce spočítáme $f(t)$ pro $t = \{t \in \mathbb{N}; t \in \langle 1, 25\rangle\}$

$$x(t) = \mathit{polyfit}(t, x, 4) = a_3t^3 + a_2t^2 + a_1t + a_0,$$
$$y(t) = \mathit{polyfit}(t, y, 4) = b_3t^3 + b_2t^2 + b_1t + b_0.$$

Následně spočítáme první a druhou derivaci polynomu

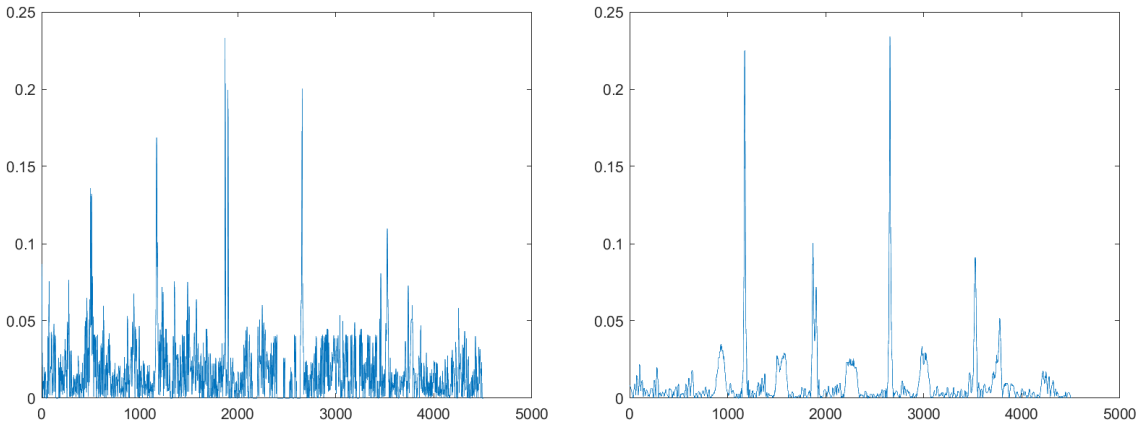
$$x'(t) = 3a_3t^2 + 2a_2t + a_1,$$
$$y'(t) = 3b_3t^2 + 2b_2t + b_1,$$
$$x''(t) = 6a_3t + 2a_2,$$
$$y''(t) = 6b_3t + 2b_2.$$

V každém i-tém kroku algoritmu (vyjma prvního a posledního) tímto způsobem nalezneme příslušný polynom $f(t) = (x(t), y(t))$, $t = \{t \in \mathbb{N}; t \in \langle 1, 25\rangle\}$. Pak pro každé $t_0 \in \langle 10, 18\rangle \subset \mathbb{N}$ vypočítáme křivost

$$\kappa(t_0) = \frac{x'(t_0)y''(t_0) + y'(t_0)x''(t_0)}{((x'(t_0))^2 + (y'(t_0))^2)^{\frac{3}{2}}}$$

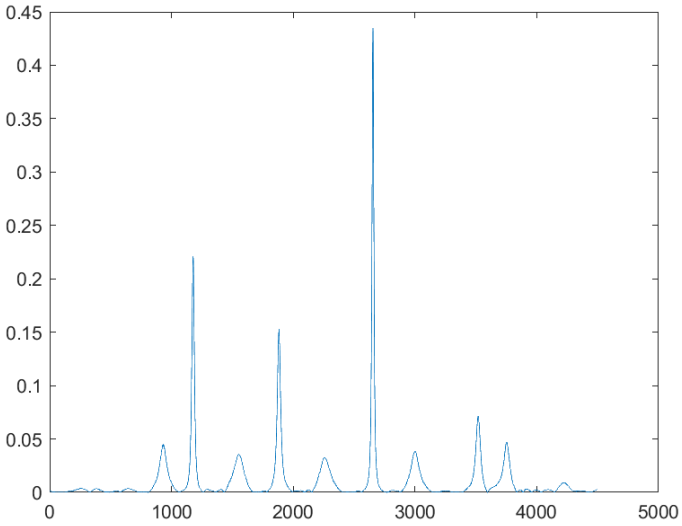
a zapíšeme ji do vektoru K na stejnou pozici, jako se počítaný pixel nachází ve vektoru $vektorHVS$.

Proměnné, které jsme zavedli na začátku kapitoly, musíme ještě upravit. Pokud bychom vytvářeli aproximační křivku pouze z devíti pixelů, aproximace by byly daleko přesnější a odpovídaly tak více pixelům hranice, což je nežádoucí. Nalezená hranice není moc hladká a náhodné vybočující pixely jsou zavádějící. Navíc devět pixelů je v poměru s celou hranicí nevýznamných a na takové malé části je hranice z celkového pohledu vždy rovná. Musíme použít počet $delkaKr$ takový, abychom na tomto počtu pixelů zaznamenali právě tu křivost hranice, kterou hledáme, tedy celé špičky a úpatí prstů i s rovnou části za nimi. Na obrázku 4.31 jsou znázorněné grafy křivostí hranic při použití počtu pixelů k aproximaci $delkaKr = 9$ a $delkaKr = 25$. Použijeme-li těchto pixelů 91 ($delkaKr = 91$), z grafu křivostí bude možné vyčíst body zlomu, které hledáme (viz obrázek 4.32). Další proměnné necháme tak, jak jsme je zavedli dříve, tj. $krok = 3$ a $pocetB = 3$. Malým počtem $pocetB$ zvyšujeme přesnost počítaných křivostí, jelikož pro výpočet křivosti každého pixelu bereme v úvahu velké okolí. A $krok = 3$ zaručuje, že aproximace bude odpovídat tvaru hranice.



(a) $delkaKr = 9$ (b) $delkaKr = 25$

Obrázek 4.31: Špatné nastavení proměnných při výpočtu křivosti hranice

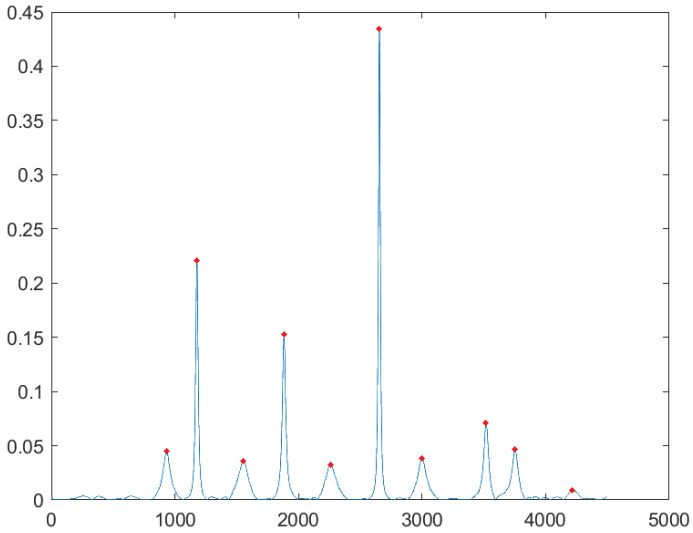


Obrázek 4.32: Křivost hranice

Body zlomu

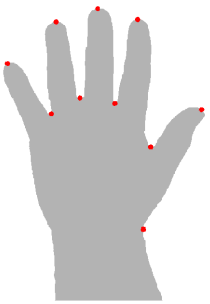
Z výsledného grafu křivostí nyní vyčteme a definujeme pixely hranice s maximální křivostí. Existuje deset viditelných zlomů, které můžeme pozorovat jak na ruce, tak i v grafu na obrázku 4.32. Jedná se o pět špiček prstů, čtyři jejich úpatí a jeden méně pozorovatelný zlom zápěstí pod palcovou hranou. V grafu tedy hledáme deset největších vrcholů, přičemž víme, že křivosti jsme zaznamenávali do vektoru K v postupném pořadí pixelů hranice z levé strany objektu. Hodnota v grafu $x = 1$ tedy odpovídá prvnímu pixelu hranice (malíková hrana) a hodnota $x = 4501$ odpovídá poslednímu pixelu hranice (palcová hrana).

Dle grafického znázornění křivostí a pozorovaného počtu pixelů na jeden vrchol můžeme určit, že o lokální maximum se jedná v případě, že v okolí 60 pixelů se nenachází jiný pixel s vyšší křivostí. Proto ve vektoru K najdeme pixely, které splňují podmínku, že šedesát pixelů v pořadí před a šedesát pixelů v pořadí po nemá vyšší hodnotu křivosti. Z těchto pixelů pak vybereme 10, které mají maximální křivost.



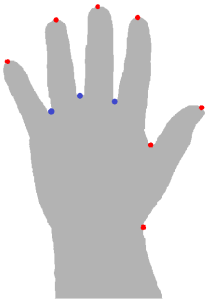
Obrázek 4.33: Maximální křivosti pixelů hranice

Nyní máme deset bodů maximálních křivostí (bodů zlomu), které jsou znázorněné na obrázku 4.34. Abychom byli však schopní rozdělit objekt ruky na objekty jednotlivých prstů a dlaně, potřebujeme znát další tři body, díky nimž dodefinujeme oříznutí malíčku, ukazováčku a palce. Jedná se o body vnějších stran prstů.



Obrázek 4.34: Body zlomu - body maximálních křivostí

K nalezení dalších bodů zlomu využijeme geometrie objektu ruky. Můžeme si všimnout, že body v úpatí prstů malíčku, prsteníčku, prostředníčku a ukazováčku s dvěma vnějšími zlomovými body, který hledáme, často leží na jedné kružnici. Proto z těchto tří bodů (viz obrázek 4.35) zkonstruujeme kružnici.



Obrázek 4.35: Hledání bodů zlomu - body pro kružnici

Stejně jako v předchozí části kapitoly označíme vodorovnou pozici pixelu v matici jako x-ovou souřadnici a svislou pozici pixelu jako y-ovou souřadnici.
Hledáme rovnici kružnice v jejím středového tvaru

$$(x-m)^2+(y-n)^2=r^2,$$

kde $S=[m,n]$ je střed kružnice. Do této rovnice dosadíme souřadnice našich tří bodů $[x_1,y_1],[x_2,y_2],[x_3,y_3]$.

$$\begin{aligned}x_1^2-2x_1m+m^2+y_1^2-2y_1n+n^2&=r^2\\x_2^2-2x_2m+m^2+y_2^2-2y_2n+n^2&=r^2\\x_3^2-2x_3m+m^2+y_3^2-2y_3n+n^2&=r^2\end{aligned}$$

Rovnice od sebe odečteme a získáme dvě rovnice o dvou neznámých.

$$\begin{aligned}(-2x_1+2x_2)m+(-2y_1+2y_2)n&=-(x_1^2-x_2^2)-(y_1^2-y_2^2)\\(-2x_1+2x_3)m+(-2y_1+2y_3)n&=-(x_1^2-x_3^2)-(y_1^2-y_3^2)\end{aligned}$$

Ty pak můžeme zapsat v maticovém tvaru a vyjádřit neznámé m a n.

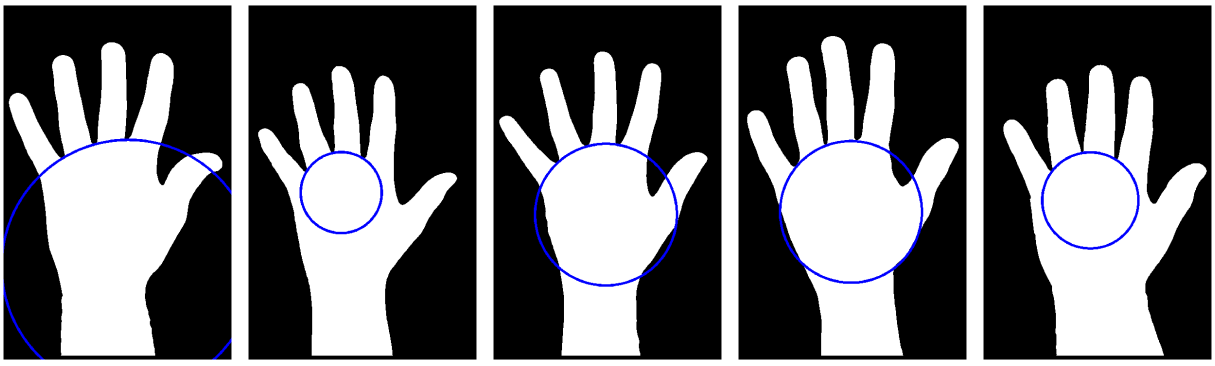
$$\begin{bmatrix}(-2x_1+2x_2) & (-2y_1+2y_2)\\(-2x_1+2x_3) & (-2y_1+2y_3)\end{bmatrix}\cdot\begin{bmatrix}m\\n\end{bmatrix}=\begin{bmatrix}-(x_1^2-x_2^2)-(y_1^2-y_2^2)\\-(x_1^2-x_3^2)-(y_1^2-y_3^2)\end{bmatrix}$$

$$\begin{aligned}A\cdot X&=B\\X&=A^{-1}\cdot B\end{aligned}$$

Poloměr kružnice pak spočítáme dosazením souřadnic středu m, n do rovnice kružnice

$$r=\sqrt{(x-m)^2+(y-n)^2}.$$

Avšak některých případech kružnice neprojde vnější stranou prstů, protože body, z kterých vytváříme kružnici, mají větší rozdíl v y-ové souřadnici a vytvoří tak malou kružnici (viz příklady na obrázku 4.36).

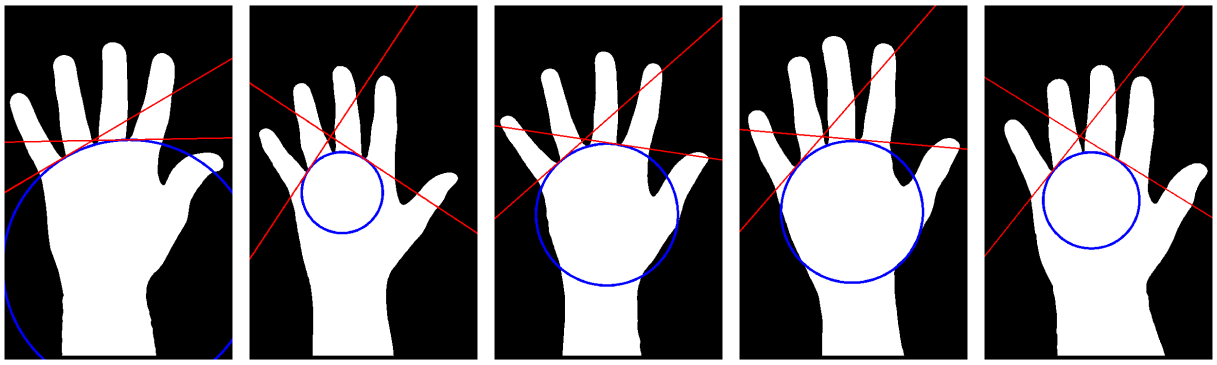


Obrázek 4.36: Hledání bodů zlomů - kružnice

Abychom vyhověli všem možným případům, potřebujeme zajistit, abychom body zlomu hledali trochu jiným způsobem. Proto po nalezení kružnice vytvoříme její tečny v krajních bodech zlomu. Rovnice

$$(x_0-m)(x-m)+(y_0-n)(y-n)=r^2$$

je rovnicí tečny kružnice se středem v bodě $S = [m, n]$ a poloměrem r v bodě $[x_0, y_0]$. Dosazením bodů $[x_1, y_1]$ a $[x_3, y_3]$ získáme dvě tečny, které v místě protnutí s hranicí vytváří další dva body zlomu (obrázek 4.37).



Obrázek 4.37: Hledání bodů zlomů - tečny kružnice

Jelikož se jedná o pixely v obrazové matici, pak jejich dosazení do výše počítané kružnice a tečen musíme zaokrouhlit. Čím více hodnoty zaokrouhlíme, tím více budou křivky širší. Děláme to proto, aby výsledná počítaná křivka byla v pixelech minimálně 4-spojitelná. Kdyby byla 8-spojitelná, nemusí se potkat s 8-spojitelnou hranicí, a pak bychom neurčili jejich průsečík. Proto raději zvolíme širší objekt reprezentující křivku, která protne hranici ve více pixelech a z nichž pak vybereme prostřední pixel.

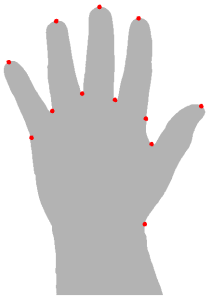
O skupinu pixelů z nichž určíme bod zlomu na malíkové hraně se jedná, když:

- nachází se vlevo dole od bodu zlomu $[x_1, y_1]$
- je od bodu $[x_1, y_1]$ vzdálen víc jak 1/3 šířky prstu (např. prsteníčku)
- jde o hraniční pixel z matice HVS
- jde o pixel tečny (případně jde o pixel kružnice, jestliže je průsečík tečny s hranicí nad úrovní bodu zlomu $[x_1, y_1]$)

O skupinu pixelů z nichž určíme bod zlomu na ukazovákové hraně se jedná, když:

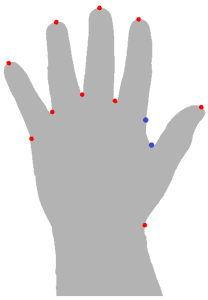
- nachází se vpravo dole od bodu zlomu $[x_3, y_3]$
- je od bodu $[x_3, y_3]$ vzdálen víc jak 1/3 šířky prstu (např. prsteníčku)
- jde o hraniční pixel z matice HVS
- jde o pixel tečny (případně jde o pixel kružnice, jestliže je průsečík tečny s hranicí nad úrovní bodu zlomu $[x_3, y_3]$)
- nenachází se mezi vyřazenými pixely, viz poznámka 4.2.2

Poznámka 4.2.2. Jestliže se tečna nachází příliš vysoko (nad úrovní bodu zlomu $[x_3, y_3]$) a volíme pak průsečík s kružnicí, nastává problém, že skupin průsečíků s kružnicí ve vybraném prostoru ukazováčku nalezneme víc, jelikož se na této straně nachází palec. Z podezřelých pixelů pak potřebujeme vyřadit pixely, který neprotínají pouze ukazovákovou hranu. Proto u každého podezřelého pixelu vypočítáme vzdálenost od bodu zlomu $[x_3, y_3]$. Pak najdeme tu nejkratší a všechny pixely vzdálenější než tato vzdálenost $+20mm$ vyřadíme.



Obrázek 4.38: Nalezené body zlomu kružnicí a tečnami

Posledním bodem zlomu, který hledáme, je bod na vnější straně palce. Proložíme-li přímkou nově získaný bod vnější strany ukazováčku a bod úpatí palce (viz obrázek 4.39), tento hledaný bod zlomu bude ležet na průsečíku zkonstruované přímky s vnější hranicí palce.



Obrázek 4.39: Hledání bodů zlomu - body pro přímku

Do obecné rovnice přímky $ax + by + c = 0$ chceme dosadit body $[x_4, y_4]$ a $[x_5, y_5]$, přičemž bod, který se nachází výš označíme jako $[x_4, y_4]$ a bod nacházející se níž jako $[x_5, y_5]$. Přímka bude tedy směřovat směrem dolů a její směrový vektor bude roven

$$\vec{u} = (x_5 - x_4, y_5 - y_4).$$

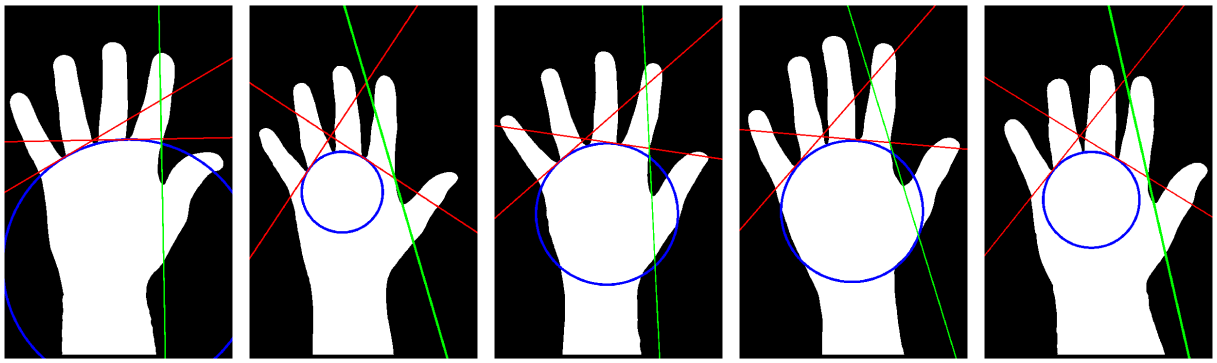
Pak normálový vektor

$$\vec{n} = (-(y_5 - y_4), x_5 - x_4)$$

obsahuje koeficienty a a b ($\vec{n} = [a, b]$). Poslední koeficient c spočítáme dosazením a, b a jednoho z bodů do rovnice přímky. Tedy

$$c = -ax_4 - by_4.$$

I při vykreslování této přímky zaokrouhlujeme a získáme tak vícepixelový objekt reprezentující křivku (obrázek 4.40).

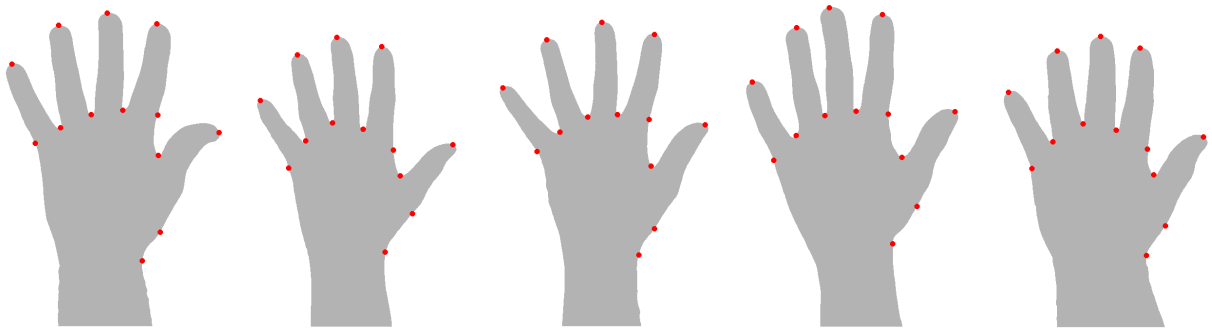


Obrázek 4.40: Hledání bodů zlomů - přímka

Nyní nalezneme skupinu pixelů, které odpovídají hledanému průsečíku s hranicí. Z této skupiny pak vybereme prostřední pixel.
O skupinu pixelů z nichž určíme bod zlomu na palcové hraně se jedná, když:

- nachází se pod rovinou bodu zlomu $[x_5, y_5]$
- je od bodu $[x_5, y_5]$ vzdálen víc jak 1/3 šířky prstu (např. prsteníčku)
- jde o hraniční pixel z matice HVS
- jde o pixel přímky
- nenachází se mezi vyřazenými pixely, viz poznámka 4.2.3

Poznámka 4.2.3. Pokud nastane situace, že x-ová souřadnice bodu $[x_4, y_4]$ je větší než x-ová souřadnice bodu $[x_5, y_5]$, pak by přímka mohla znovu protnou hranici v oblasti předloktí. Proto i v tomto případě ošetříme hledaní skupiny podezřelých bodů. Všechny pixely vzdálené více jak nejbližší průsečík $+20mm$ vyřadíme z této skupiny.



Obrázek 4.41: Výsledné body zlomu

4.2.2. Rozdělení objektu

V této kapitole se pokusíme získat jednotlivé části objektu ruky. Naším cílem bude oddělit prsty od dlaně pomocí bodů zlomu a získat samostatnou dlaň, která bude ukončena v zápěstí, tedy objekt dlaně nebude obsahovat předloktí.

U prstů nás nebude zajímat pouze definice samotného objektu, ale i 8-spojitelná křivka, která tvoří přesný střed prstu. Díky ní pak dokážeme spočítat jejich povrch.

Začneme postupnou erozí celého objektu, tzn. v každém kroku i odebereme hraniční pixely z matice VYS (matice výsledného objektu ruky). Zároveň budeme do matice ER na pozice odebraných pixelů zapisovat hodnoty i, tj. číslo kroku, v kterém byly pixely odebrány. Otázkou nyní je, jestli v postupné erozi budeme za hraniční pixely považovat pixely hraniční z hlediska okolí $\mathcal{N}_4(r,s)$, nebo okolí $\mathcal{N}_8(r,s)$. Podívejme se na barevné zobrazení matice ER (obrázek 4.42), kde hodnoty pixelů $i = 1$ až $i =$poslední krok eroze jsou znázorněný bílou až černou barvou.

(a) okolí $\mathcal{N}_4(r,s)$ (b) okolí $\mathcal{N}_8(r,s)$ (c) střídavé okolí

Obrázek 4.42: Postupná eroze hraničních pixelů z hlediska použitého okolí

Z obrázků 4.42a a 4.42b je vidět, že při použití okolí $\mathcal{N}_4(r,s)$ a $\mathcal{N}_8(r,s)$ se v každém kroku eroze zachovává a spíše zvýrazňuje ostrý tvar hranice, což je pro budoucí kroky nevýhodné až nepřijatelné. Navíc víme, že pokud použijeme k erozi okolí $\mathcal{N}_4(r,s)$, výsledné hraniční pixely budou 8-spojitelné a naopak (viz poznámka 3.4.2). Proto je nejlepší variantou výběr hraničních pixelů střídat. V i-tém kroku považujeme za hraniční pixely takové pixely, které mají v svém okolí $\mathcal{N}_4(r,s)$ méně nebo právě tři objektové pixely. Potom v $i+1$ kroku označíme pixely za hraniční, pokud ve svém okolí $\mathcal{N}_8(r,s)$ mají méně nebo právě sedm objektových pixelů. Tímto způsobem získáme matici ER, kterou můžeme vidět barevně zobrazenou na obrázku 4.42c.

Nyní můžeme z matice ER najít středové křivky prstů. Tyto křivky není tak jednoduché najít a musíme provést několik kroků, které budou v každém případě zaručovat správné výsledky.

1. Vytvoříme binární matici SE, která bude obsahovat středové křivky prstů. Budeme procházet matici ER a každý nenulový pixel, který ve svém kruhovém okolí o poloměru 5 pixelů nemá pixel s vyšší hodnotou, zapíšeme do matice SE. Tímto způsobem vybereme lokální maxima prstů a dlaně.

2. Každé dva body zlomu v úpatí každého prstu propojíme přímkou stejným způsobem jako v kapitole 4.1.2 (viz obrázek 4.13 a rovnice 4.1). Na této spojnici najdeme pixel s největší hodnotou eroze v matice ER, což označíme jako patu prstu. Špička prstu je nám známa, protože je to konkrétní bod zlomu.

3. Všechny pixely v matici SE, které nenáleží prstům, odstraníme (přepíšeme jejich hodnotu na nulu). Vyskytují se v prostoru: [maximální svislá pozice z pat malíčku až ukazováčku : spodní okraj obrazu; levý okraj obrazu : vodorovná pozice paty palce].

4. Nalezneme hodnotu $maxVZD$, která bude definovat maximální možnou vzdálenost pixelů, které patří jednomu prstu. Tato hodnota se bude rovnat minimální vzdálenosti ze všech vzdáleností pat prstů.

5. Ke každému pixelu takovému, že $SE(p_1(1), p_1(2)) = 1$, spočítáme vzdálenosti ke všem ostatním pixelům $SE(r, s) = 1$. Vybereme pixel p_2, který má s pixelem p_1 nejmenší vzdálenost a pokud bude tato vzdálenost menší než $5/6 \cdot maxVZD$, pak tyto dva pixely spojíme přímkou. Propojíme tak části křivky středů prstů získané předchozími kroky a podmínkou $5/6 \cdot maxVZD$ ošetříme, aby se nepropojily prsty navzájem.

6. Dilatací a následnou erozí s kruhovým okolím o poloměru 5 pixelů křivky trochu vyhladíme, aby například neobsahovaly tolik vyčnívajících pixelů.

7. Nyní všechny nalezené středové pixely každého prstu srovnáme a zapíšeme je do vektoru vzd v pořadí od špičky po patu prstu. Pro každý prst provedeme následující algoritmus popsaný na obrázku 4.43, který vychází z matice SEp. Tato matice je na začátku algoritmu kopií matice SE, ale v jeho průběhu z ní postupně mažeme (tj. přepisujeme hodnotu jedna na hodnotu nula) pixely, které jsme již zapsali do vektoru vzd. Pixel p bude v každém i-tém kroku pixelem naposledy zapsaným do vzd a pixelem o značíme konkrétní pixel v jeho okolí $\mathcal{N}_8^p(r, s)$. Značení těchto okolních pixelů vychází z obrázku 4.21.

8. Pro každý prst spočítáme vzdálenost každého pixelu od špičky prstu.

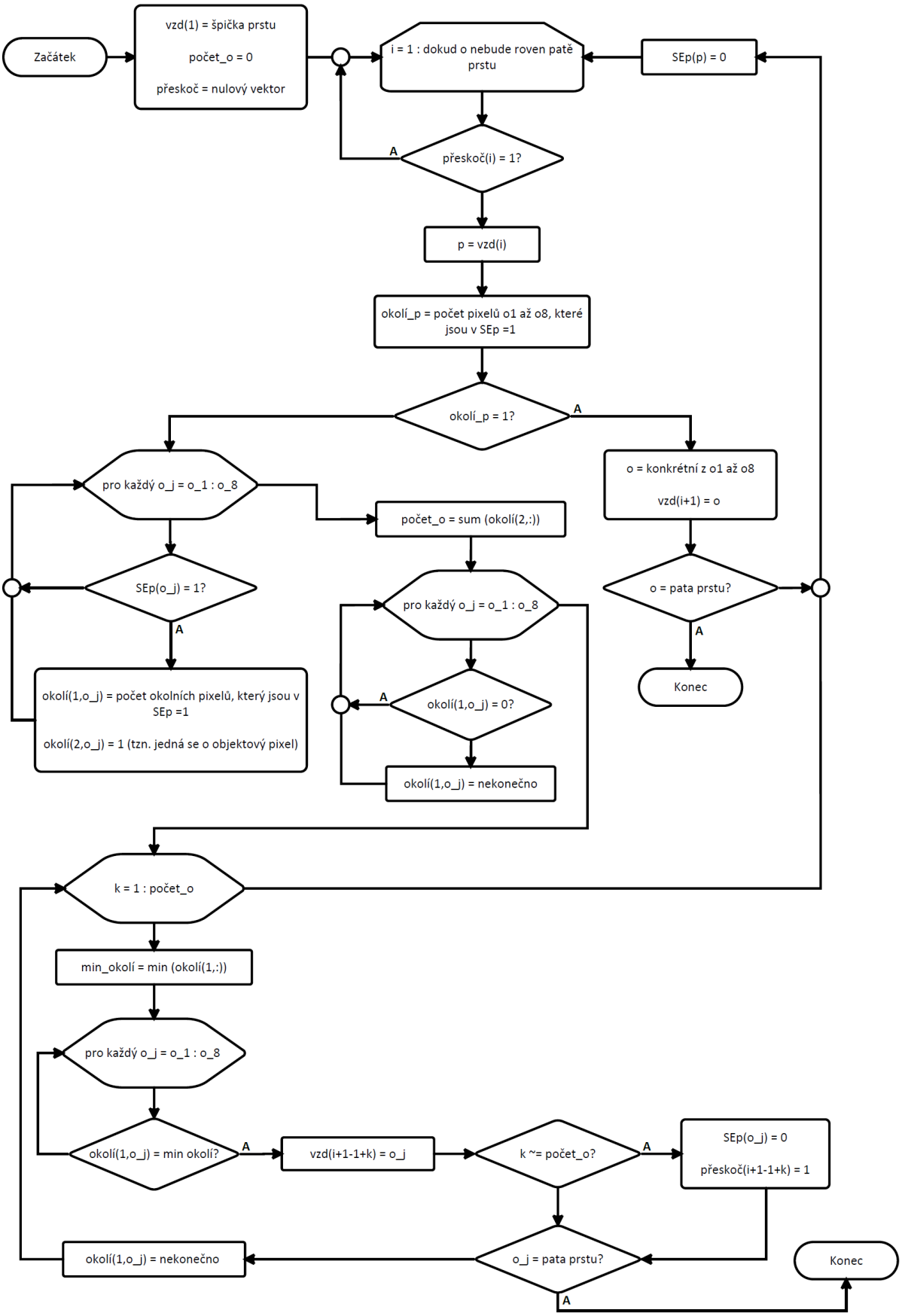
9. Díky předchozím vzdálenostem můžeme nyní opravit chybné posloupnosti pixelů ve vektoru vzd. Mějme posloupnost pixelů s indexy: $i-2, i-1, i, i+1$. Označme v jako vzdálenost pixelu od špičky prstu, y jako svislou pozici pixelu v matici SE a x jako pozici vodorovnou. Jestliže platí

$$v(i-1) > v(i)$$

a zároveň platí

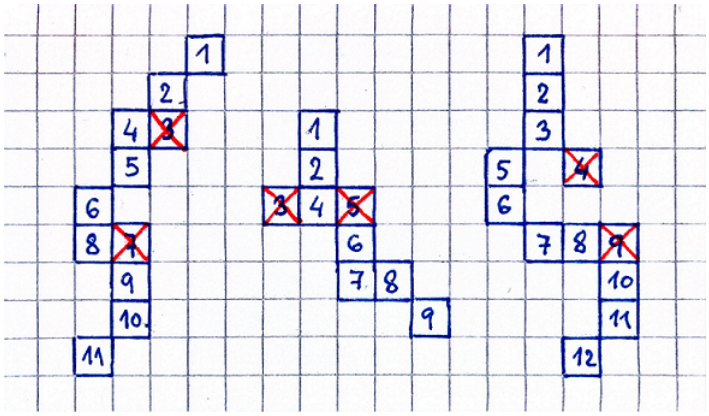
$$\begin{aligned} &|y(i) - y(i-2)| \leq 1 \quad \wedge \quad |x(i) - x(i-2)| \leq 1 \quad \wedge \\ &|y(i-1) - y(i+1)| \leq 1 \quad \wedge \quad |x(i-1) - x(i+1)| \leq 1, \end{aligned} \tag{4.3}$$

pak pixely s indexy $i-1$ a i ve vektoru vzd vyměníme. Podmínkami 4.3 ověřujeme, že i při výměně pixelů na sebe budou vedlejší pixely navazovat.



Obrázek 4.43: Vývojový diagram algoritmu pro řazení pixelů středových křivek

10. Ze středových křivek prstů vytvoříme křivky 8-spojitelné. Pro každý i-tý pixel z vektoru vzd budeme sledovat okolí $\mathcal{N}_8(r,s)$ $(i+1)$-tého pixelu. Pokud se v tomto okolí nachází $(i+2)$-tý pixel a zároveň $(i-1)$-tý pixel, pak pixel na i-té pozici je zbytečný a z vektoru vzd ho odstraníme. Další pixely, které touto úpravou odstraníme, jsou takové, které ve svém okolí nemají předchozí $(i-1)$ a zároveň následující pixel $(i+1)$. Tyto úpravy jsou lépe vidět na obrázku 4.44.



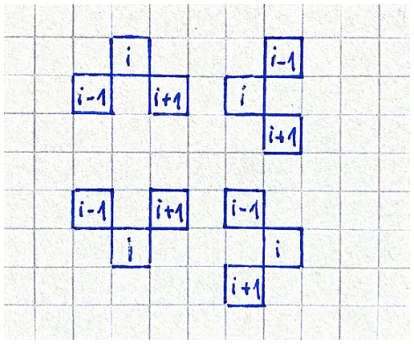
Obrázek 4.44: Příklady pixelů nepatřících do 8-spojitelných křivek středů prstů

11. Z předchozího odstranění řádků se ve vektoru vzd objevují nulové řádky. Proto vytvoříme nový vektor $vzdV$, který bude obsahovat přeindexovaný vektor vzd.

12. Následně znovu provedeme krok 10. Jediným rozdílem bude použití vektoru $vzdV$ namísto vektoru vzd. Musíme ho provést znovu, protože existují případy, kdy v původním vektoru vzd jsou dva pixely narušující 8-spojitelnost hned za sebou a krokem 10 odstraníme pouze první špatný pixel.

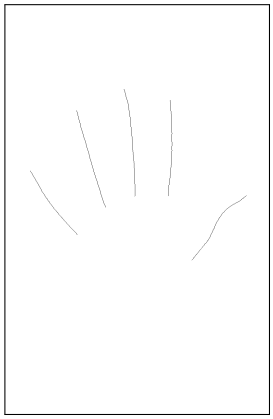
13. Opětovným přeindexováním získáme 8-spojitelné křivky středů prstů ve správném pořadí od špičky až k patě prstu.

14. Posledním krokem provedeme korekci křivek. Nalezneme pixely stejného typu jako na obrázku 4.45 a zarovnáme je na vodorovnou či svislou úroveň mezi předchozí pixel $i-1$ a následující pixel $i+1$.



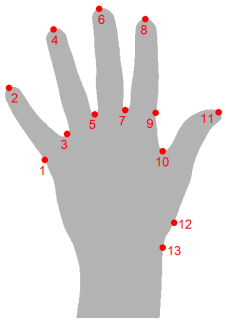
Obrázek 4.45: Případy pixelů ke korekci křivek středů prstů

Těmito kroky získáme křivky středů prstů zobrazené na obrázku 4.46.



Obrázek 4.46: Křivky středů prstů (negativ)

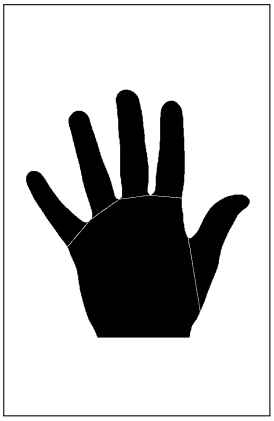
Následně definujeme dlaň ruky. Pro jednodušší popis jak dlaň získáme, označíme body zlomu (viz obrázek 4.47).



Obrázek 4.47: Označení bodů zlomu

Než začneme definovat objekt dlaně, nejprve zjednodušíme hranici tak, aby neobsahovala hraniční pixely stejného typu jako 3. pixely na obrázku 4.16. Tyto pixely se odlišují od jiných hraničních pixelů tím, že v jejich okolí $\mathcal{N}_8(r,s)$ se nachází přesně tři objektové pixely. Kromě tohoto typu pixelů, existuje ještě jeden typ, který má ve své osmisousednosti přesně tři pixely. Jedná se o vyčnívající pixely z objektu, který ve svém $\mathcal{N}_4(r,s)$ okolí mají pouze jeden další objektový pixel. Tyto pixely jsme však odstranili při definování výsledného objektu ruky zapsaného do matice VYS. Vytvoříme tedy matici objektu se zjednodušenou hranicí $VYSz$ tak, že z matice VYS odstraníme hraniční pixely, který splňují podmínku, že v jejich $\mathcal{N}_8(r,s)$ se nachází tři objektové pixely. Díky zjednodušené hranici můžeme použít následující algoritmus hledající hranici dlaně bez toho, abychom museli ošetřovat tyto případy pixelů.

Tvar dlaně definujeme pomocí bodů zlomu. Propojíme dvojice bodů zlomu v úpatí každého prstu přímkou (body $1-3$, $3-5$, $5-7$, $7-9$, $10-12$) a každému pixelu na ní ležícím udělíme hodnotu nula. Pomocí bodu zlomu 13 definujeme místo zápěstí a oddělíme tu dlaň od předloktí, které odstraníme.



Obrázek 4.48: Oddělení dlaně (negativ)

V matici s rozdělenými objekty (obrázek 4.48) najdeme všechny hraniční pixely a zapíšeme je do matice HD. Jsou to takové pixely, které jsou objektové a v okolí $\mathcal{N}_4(r,s)$ mají jeden až tři jiné objektové pixely. Z těchto hraničních pixelů však chceme získat pouze hraniční pixely objektu dlaně.

Následující algoritmus pro hledání hranice dlaně bude místy podobný Algoritmu pro nalezení hranice objektu v 4.1.2. Avšak rozdílnosti zde jsou a proto se nemůžeme na tento algoritmus jednoduše odkázat.

Algoritmus pro nalezení hranice dlaně

Jelikož dlaň a prsty nyní dělí 8-spojitelná přímka, hraniční pixely objektů jsou sousední z hlediska okolí $\mathcal{N}_D(r,s)$. V těchto místech musíme být opatrní, abychom zůstali stále na hranici dlaně. Proto zavedeme tzv. kontrolní bod $bodK$. Je to bod, který umístíme do středu zápěstí a který nám určí, kterým dalším hraničním pixelem se máme dát, budeme-li mít dvě možnosti. Jelikož se tento bod nachází uprostřed spodní části dlaně, můžeme říct, že v případě dvou sousedních hraničních pixelů z nichž jeden patří dlani a druhý dlani nepatří, je pixel patřící dlani vždy blíž ke kontrolnímu bodu.

Pixely vyhledáváme v matici hranicí HD a postupně zapisujeme do matice D a vektoru bh. První pixel hranice bude ležet na stejném řádku jako leží 13. bod zlomu, pouze na opačné straně zápěstí. Na tom stejném řádku definujeme i $bodK$, který se bude nacházet přímo mezi nimi.

Prvně nalezneme druhý a třetí pixel hranice. Druhý hledáme v okolí prvního pixelu na pozicích $o1$, $o2$ a $o3$ dle označení okolních pixelů na obrázku 4.21. Ten, který bude mít v matici HD hodnotu jedna, zapíšeme jako druhý pixel hranice. Třetí pixel nalezneme obdobně s tím rozdílem, že procházíme okolní pixely $o1$, $o2$, $o3$, $o4$ a $o8$ druhého hraničního pixelu.

Poznámka 4.2.4. Zápisem nově nalezeného pixelu vybraného z okolních pixelů $o1 - o8$ pixelu p v i-tém kroku se zde myslí:

- na nový $(i+1)$-tý řádek vektoru bh zapíšeme pozici pixelu o
- zapíšeme pixel o do matice D (tj. na jeho pozici zapíšeme hodnotu jedna)
- z matice HD odstraníme pixel o (na jeho pozici zapíšeme hodnotu nula, abychom se při dalším výběru hraničního pixelu nevraceli zpět, resp. nemuseli tuto možnost ošetřovat)

Poznámka 4.2.5. Následující popis vždy předpokládá, že posledním nalezeným pixelem v matici HD je pixel p a v jeho okolí se nachází pouze možné, ještě nenalezené pixely hranice o. Pixely již nalezené se v matici HD nenachází.

Než popíšeme algoritmus je nutné si uvědomit fakta:

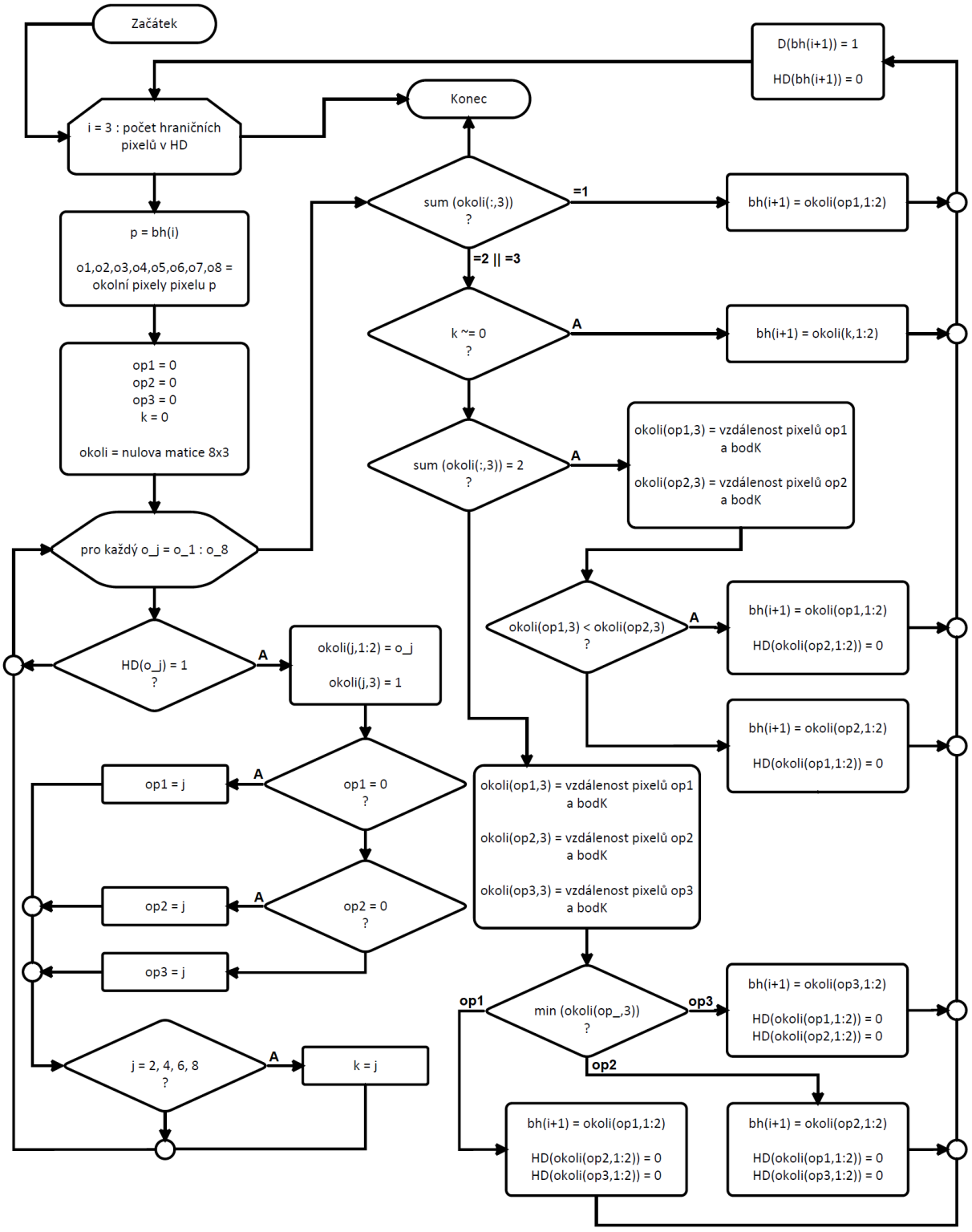
1. Jak už bylo řečeno, přímka rozdělující prsty od dlaně je 8-spojitelná, tedy hraniční pixely jiných objektů se buď nenachází ve vzájemném okolí, nebo se nachází maximálně v okolí $\mathcal{N}_D(r,s)$. Tzn. nachází-li se okolní pixel o pixelu p v jeho okolí $\mathcal{N}_4^p(r,s)$, pak pixel o náleží stejnému objektu jako náleží pixel p. Pixel v čtyřsousednosti má tedy vždy přednost.

2. Při výběru následujícího pixelu můžou nastat tři možnosti.

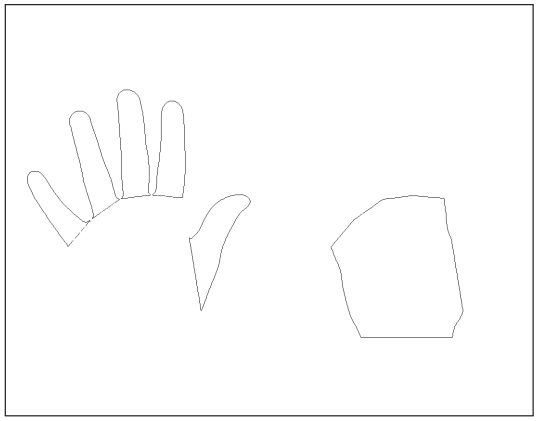
   - V okolí pixelu p nalezneme pouze jeden pixel o.
   - Nalezneme více možných pixelů o, přičemž budou maximálně tři.
     V případě tří se jedná buď o dva pixely náležící hranici dlani a jeden cizí pixel. Pak je jeden z pixelů dlaně 4-spojitelný a pokračujeme jím. Nebo se jedná o jeden náležící dlani a dva cizí. V tomto případě se všechny tři pixely nacházejí v $\mathcal{N}_D^p(r,s)$ okolí pixelu p a my hledáme ten nejbližší kontrolnímu bodu $bodK$. Jestliže nalezneme dva, zvolíme ten který se nachází v $\mathcal{N}_4^p(r,s)$ okolí nebo pak ten, který je blíž k bodu $bodK$.
   - V okolí pixelu p se nenachází žádný pixel. Tato možnost nastává v případě, že jsme prošli celou hranici dlaně a algoritmus jí končí.

Zavedeme proměnné $op1$, $op2$, $op3$, do kterých v každém kroku zapíšeme čísla podezřelých okolních pixelů (vybíráme z $1-8$). V případě, že existuje pouze jeden možný následující pixel, pak $op1$ bude obsahovat jeho číslo a proměnné $op2$ a $op3$ budou nulové. Proměnná k říká, zda je podezřelým pixelem jeden ze 4-spojitelných pixelů. Bude tedy buď nulová nebo bude rovna jednomu z čísel $2, 4, 6$, nebo 8. Do matice $okoli(8x3)$ budeme zapisovat pozice okolních pixelů $o1 - o8$ v matici HD. Třetí sloupec bude sloužit k popisu těchto pixelů. Nejprve bude obsahovat hodnotu jedna v případě, že se jedná o pixel hraniční. V druhé části algoritmu bude vyplněn vzdáleností k bodu $bodK$, bude-li tato vzdálenost potřeba.

Po nalezení prvního až třetího pixelu hranice pokračujeme algoritmem, který nalezne všechny ostatní pixely hranice dlaně. Jeho vývojový diagram je zobrazen na obrázku 4.49.



Obrázek 4.49: Vývojový diagram algoritmu pro nalezení hranice dlaně



Obrázek 4.50: Matice hranicí HD a D po provedení algoritmu z obrázku 4.49 (negativ)

Nakonec výslednou hranici dlaně v matici D vyplníme, abychom získali celý objekt, který potřebujeme pro výpočet její plochy.

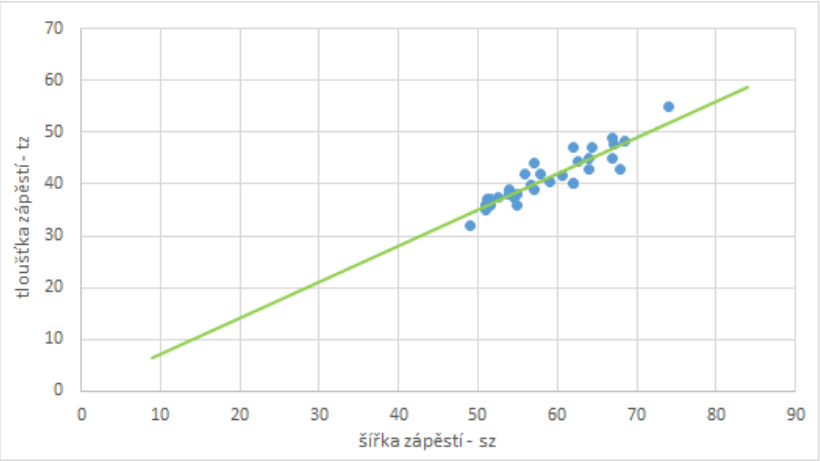
4.3. Výpočet povrchu ruky a statistické ověření výsledků

Nyní nám už nic nebrání v samotném výpočtu povrchu. Ze vstupní fotografie jsme získali všechny potřebné informace, tedy v této kapitole popíšeme navržený model výpočtu povrchu a statisticky ověříme jeho kvalitu. Pro následující výpočet potřebujeme znát tloušťku zápěstí, kterou z 2D pohledu ruky nijak nevyčteme. Tuto proměnnou musíme získat nějakým jiným způsobem. Naštěstí mezi šířkou a tloušťkou zápěstí existuje určitá závislost.

4.3.1. Odhad tloušťky zápěstí

Šířku zápěstí odečteme z matice D, v které je uložen objekt dlaně (viz obrázek 4.50 obsahující hranici tohoto objektu). 13. bod zlomu (viz obrázek 4.47) definuje místo zápěstí a jeho souřadnice jsou uložené ve vektoru $bodyZ$. Jestliže sečteme objektové pixely na řádku 13. bodu zlomu, získáme šířku zápěstí sz.

Pro odhad tloušťky zápěstí bylo provedeno 34 pokusů, tj. $n = 34$. Výsledky těchto pokusů jsou zobrazeny na obrázku 4.51.



Obrázek 4.51: Grafické zobrazení pokusů pro závislost šířky a tloušťky zápěstí

Již z grafu je závislost viditelná, avšak pomocí korelační analýzy ji ještě ověříme. Jelikož nemůžeme předpokládat normální rozdělení pravděpodobnosti těchto dat, použijeme Spearmanův korelační koeficient. Označíme šířku zápěstí jako sz a tloušťku zápěstí jako tz. Naměřená data srovnáme vzestupně podle hodnot sz. Pak sloupec R_i odpovídá pořadí hodnot sz a sloupec Q_i odpovídá pořadí hodnot tz.

$\boldsymbol{sz}$:	$\boldsymbol{tz}$:	$\boldsymbol{R_i}$:	$\boldsymbol{Q_i}$:
49	32	1	1
51	35	3	2
51	36	3	4,5
51	36	3	4,5
51,1	37	5	7,5
51,5	36	6,5	4,5
51,5	37	6,5	7,5
52,5	37,5	8	9,5
54	38	10	12
54	38	10	12
54	39	10	14,5
54,5	37,5	12	9,5
55	36	13,5	4,5
55	38	13,5	12
56	42	15	21
56,6	39,7	16	16
57	39	17,5	14,5
57	44	17,5	25
57,85	42,01	19	22
59	40,5	20	19
60,6	41,55	21	20
62	40	23	17,5
62	40	23	17,5
62	47	23	29,5
62,6	44,4	25	26
64	43	26,5	23,5
64	45	26,5	27,5
64,4	47	28	29,5
67	45	29,5	27,5
67	49	29,5	33
67,08	47,62	31	31
68	43	32	23,5
68,55	48,25	33	32
74	55	34	34

Pak Spearmanův korelační koeficient je roven

$$r_S = 1 - \frac{6}{34(34^2-1)} \sum_{i=1}^{34} (R_i - Q_i)^2 = 0,93056.$$

Jelikož je počet pozorování $n > 30$, kritické hodnoty $r_S(\alpha)$ nenajdeme mezi tabularizovanými na obrázku 2.1. Proto spočítáme $r_S^*(\alpha)$ pro $\alpha = 0,01$ a pro $\alpha = 0,05$:

$$r_S^*(0,01) = \frac{u(0,01/2)}{\sqrt{n-1}} = 0,44839, \quad r_S^*(0,05) = \frac{u(0,05/2)}{\sqrt{n-1}} = 0,34119,$$

kde kvantily normovaného normálního rozdělení $N(0,1)$ jsou $u(0,01/2) = 2,575829$, $u(0,05/2) = 1,959964$. Kvantily najdeme ve statistických tabulkách například v [2].

Jelikož platí $|r_S| > r_S^*(0,05)$ a dokonce platí i $|r_S| > r_S^*(0,01)$, zamítáme nezávislost tloušťky a šířky zápěstí na hladině významnosti $0,01$.

Nyní, za pomoci regresní analýzy vypočítáme odhad tloušťky zápěstí. Dle grafického znázornění dat (viz 4.51), kde můžeme pozorovat lineární závislost, zvolíme obecnou přímku jako regresní funkci. Tedy

$$y_i = \beta_0 + \beta_1 x_i,$$

kde nezávisle proměnnou x je šířka zápěstí sz a závisle proměnnou y je tloušťka zápěstí tz. Bodové odhady regresních koeficientů $\boldsymbol{\beta} = (\beta_0, \beta_1)$ určíme metodou nejmenších čtverců. Dosazením do vzorců pro model regresní přímky 2.14 máme

$$\bar{x} = \frac{1}{34}\sum_{i=1}^{34} x_i = 58,58176, \quad \bar{y} = \frac{1}{34}\sum_{i=1}^{34} y_i = 41,05971,$$

$$b_1 = \frac{\sum_{i=1}^{34} x_i y_i - 34\bar{x}\bar{y}}{\sum_{i=1}^{34} x_i^2 - 34\bar{x}^2} = 0,70216, \quad b_0 = \bar{y} - b_1\bar{x} = -0,07418.$$

Získali jsme tedy regresní funkci ve tvaru

$$y = -0,07418 + 0,70216x.$$

Avšak koeficient β_0 je velmi nízký a možná i statisticky nevýznamný. Proto ho budeme testovat na hypotézu $H_0 : \beta_0 = 0$. Testovací kritérium

$$t = \frac{b_0}{\sqrt{s^2 v_{11}}}$$

obsahuje bodový odhad rozptylu σ^2 a diagonální prvek matice $\boldsymbol{V} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}$, tj

$$s^2 = \frac{\sum_{i=1}^{34} y_i^2 - b_0\sum_{i=1}^{34} y_i - b_1\sum_{i=1}^{34} x_i y_i}{34-2} = 4,05486,$$

$$\boldsymbol{V} = \begin{pmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{pmatrix}^{-1} = \begin{pmatrix} 34 & 1991,78 \\ 1991,78 & 118048,7 \end{pmatrix}^{-1} = \begin{pmatrix} 2,54042 & -0,04286 \\ -0,04286 & 0,00073 \end{pmatrix}.$$

Hodnota testovacího kritéria je po dosazení rovna $t = -0,02311$ a jelikož náleží doplňku kritického oboru $\bar{W}_{0,05} = \langle -t_{34-2}(1-0,05/2); t_{34-2}(1-0,05/2)\rangle = \langle -2,037; 2,037\rangle$, tak hypotézu $H_0 : \beta_0 = 0$ nezamítáme na hladině významnosti $\alpha = 0,05$. Tedy koeficient β_0 zanedbáme a výsedný regresní model závislosti tloušťky na šířce zápěstí je

$$tz = 0,70216 \cdot sz. \tag{4.4}$$

4.3.2. Povrch ruky

Výpočet povrchu ruky rozdělíme na povrch prstů Sp a povrch dlaně Sd, který nakonec sečteme. Výsledný povrch však bude v jednotkách pixelů, proto tuto hodnotu ještě nakonec převedeme na mm^2.

Povrch prstů

Připomeňme názvy proměnných. Ve vektoru $vzdVys$ jsou pro každý prst uloženy souřadnice pixelů tvořící střed prstu od jeho špičky k jeho patě (viz obrázek 4.46). Matice ER (viz její barevné zobrazení na obrázku 4.42c) byla vytvořena postupnou erozí celého objektu ruky. Obsahuje hodnoty pixelových vzdáleností každého pixelu k hraně objektu. Tedy středové pixely prstů obsahují v matici ER poloměr prstu v dané výšce prstu.

V každém středovém pixelu p každého prstu vytvoříme válec jehož výška je rovna jednomu pixelu, tedy $v = 1$ a poloměr je jeho hodnota v matici ER, tedy $r = ER(p)$. Pak plášť tohoto válce přičteme k povrchu prstů, tj.

$$Sp = Sp + 2\pi rv.$$

Tímto způsobem projdeme všechny středové pixely všech prstů zapsané ve vektoru $vzdVys$ a získáme povrch prstů.

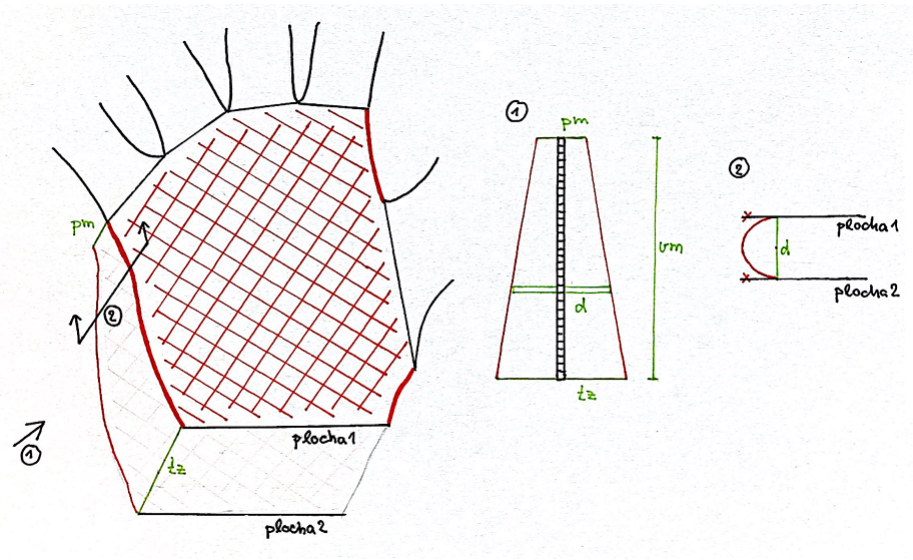
Povrch dlaně

Pro výpočet povrchu dlaně bychom mohli pouze sečíst objektové pixely v matici dlaně D a vynásobit ji dvěma. Tedy by stačilo

$$Sd = 2 \sum_r \sum_s D[r, s].$$

Ale tento odhad by byl velmi hrubý, především kvůli okrajům dlaně, které jsou ve skutečnosti zaoblené. Proto se pokusíme vytvořit korekci tohoto vzorce.

Jak už je v předchozí části kapitoly zmíněno, středové pixely prstů mají v matici ER zaznamenanou hodnotu poloměru. Tzn. že známe poloměr paty malíčku, ukazováčku a palce, tedy i jejich průměry pm, pu a pp. Následující obrázek naznačuje korekci povrchu dlaně.



Obrázek 4.52: Korekce dlaně

Označme hranu mezi malíčkem a zápěstím jako *malíkovou hranu*, hranu mezi ukazováčkem a placem jako *palcovou hranu nad palcem* a hranu mezi placem a zápěstím jako *palcovou hranu pod palcem*. Naším cílem bude na těchto hranách nahradit okraj plochy 1 a okraj plochy 2 půlkružnicí.

Jak je naznačeno na obrázku 4.52, plocha 1 a plocha 2 nemají konstantní vzdálenost. Z toho plyne, že půlkružnice má v každém místě hrany jiný průměr. Jelikož tyto průměry neznáme, potřebujeme nalézt závislost průměru půlkružnice (tj. vzdálenosti ploch) na vzdálenosti pixelu od malíčku (příp. ukazováčku).

Nechť body malíkové hrany jsou $[0, pm]$, $[vm, tz]$, kde pm je průměr malíčku, vm je délka malíkové hrany definovaná jako vzdálenost 1. bodu zlomu k levému kraji zápěstí a tz je tloušťka zápěstí. Dosazením těchto bodů do rovnice přímky $y = ax + b$ získáme závislost na malíkové hraně

$$y = \frac{tz - pm}{vm}x + pm.$$

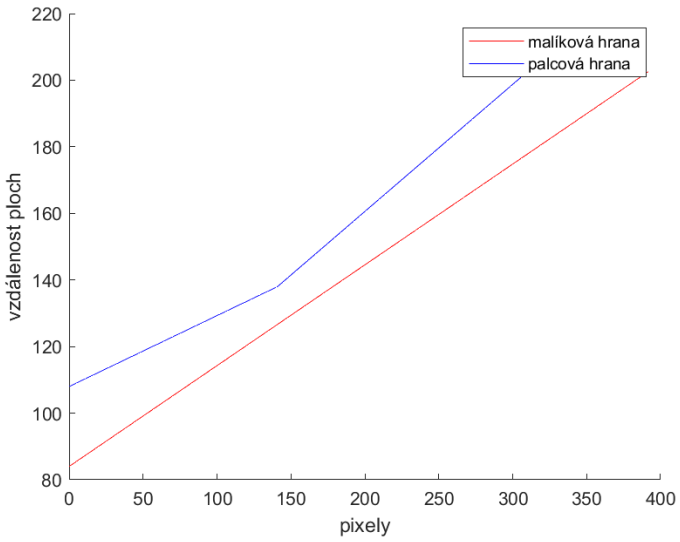
Palcová hrana se skládá ze dvou částí a pro každou z nich vyjádříme vlastní rovnici. Závislost na palcové hraně nad palcem získáme z bodů $[0, pu]$, $[vp1, pp]$, kde pu je průměr ukazováčku, $vp1$ je délka hrany nad palcem definovaná jako vzdálenost 9. a 10. bodu zlomu a pp je průměr palce. Výsledná rovnice přímky je

$$y = \frac{pp - pu}{vp1}x + pu.$$

Druhou část palcové hrany tvoří body $[vp1, pp]$, $[vp1 + vp2, tz]$, kde $vp2$ je délka hrany pod palcem definovaná jako vzdálenost 12. a 13. bodu zlomu a z kterých po dosazení do rovnice přímky dostaneme závislost

$$y = \frac{tz - pp}{vp2}x + pp - \frac{vp1(tz - pp)}{vp2}.$$

Všechny výše vyjádřené závislosti jsou zakreslené v grafu na obrázku 4.53.



Obrázek 4.53: Závislost vzdálenosti ploch dlaně na vzdálenosti pixelu od malíčku /ukazováčku

Nyní můžeme upravit původní povrch dlaně. Pro každou hranu dlaně a každý pixel délky hrany přičteme obvod půlkružnice a odečteme z plochy 1 i z plochy 2 přečnívající okraj o délce poloměru půlkružnice. Tedy pro jeden pixel z délky jedné hrany provedeme

$$Sd = Sd + \frac{\pi d}{2} - 2 \cdot \frac{d}{2},$$

kde d je aktuální průměr půlkružnice (vzdálenost ploch).

Po sečtení a převedení na mm^2, tj. $S_{model} = (Sp + Sd) \cdot 0,2^2[mm^2]$, získáme výsledný povrch ruky získaný navrženým modelem.

4.3.3. Statistické ověření vypočteného povrchu ruky

Povrch vypočtený matematickým modelem v předchozí kapitole neodpovídá reálným hodnotám povrchu. To ale nutně neznamená, že navržená metoda je špatná. Nejprve otestujeme, jestli vypočítané povrchy jsou závislé na reálných. V případě, že by byly závislé, pak stačí najít vhodný regresní model, který bude tuto závislost popisovat a získáme jím konečný odhad povrchu ruky. K dispozici máme deset pozorování ($n = 10$).

Označíme modelový povrch S_{model} jako X a reálný povrch jako Y. Závislost mezi náhodnými proměnnými X a Y budeme testovat Spearmanovým korelačním koeficientem pořadí. Hodnotám proměnných X, Y přiřadíme jejich pořadí R_i a Q_i.

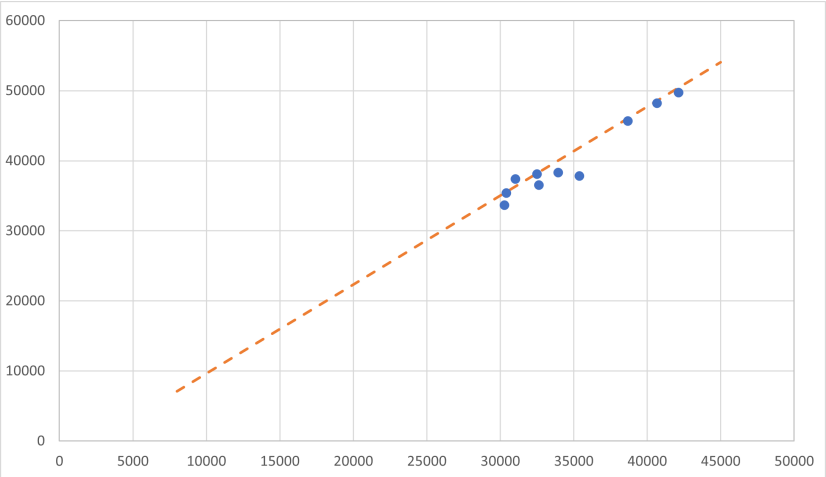
$\boldsymbol{X_i}$:	$\boldsymbol{Y_i}$:	$\boldsymbol{R_i}$:	$\boldsymbol{Q_i}$:
29470, 95	33666, 99	1	1
29704, 92	35406, 64	2	2
30161, 86	37398, 91	3	4
31831, 50	36530, 07	4	3
32064, 74	38104, 11	5	6
33161, 33	38318, 63	6	7
34482, 74	37851, 72	7	5
37829, 01	45651, 23	8	8
39993, 53	48206, 43	9	9
41283, 45	49726, 64	10	10

Testujeme hypotézu $H_0 : \rho = 0$ proti alternativní hypotéze $H_A : \rho \neq 0$. Testovací kritérium se rovná

$$r_S = 1 - \frac{6}{10(10^2 - 1)} \sum_{i=1}^{10} (R_i - Q_i)^2 = 0,95152$$

a kritické hodnoty odečteme z tabulky 2.1, tedy $r_S(0,05) = 0,6364$ a $r_S(0,01) = 0.7818$. Jelikož platí $|r_S| > r_S(0,01)$, zamítáme hypotézu o nezávislosti proměnných na hladině významnosti $0,01$.

Hodnoty proměnných X a Y vykreslíme do grafu, abychom navrhli tvar regresní funkce.



Obrázek 4.54: Grafické zobrazení pokusů pro závislost modelového a reálného povrchu

Dle rozložení dat na obrázku 4.54 použijeme regresní přímku $y_i = \beta_0 + \beta_1 x_i$, kde regresní koeficienty β_0 a β_1 odhadneme pomocí metody nejmenších čtverců. Bodové odhady b_0 a b_1 vypočteme ze vztahů

$$\bar{x} = \frac{1}{10} \sum_{i=1}^{10} x_i = 33998,402, \quad \bar{y} = \frac{1}{10} \sum_{i=1}^{10} y_i = 40086,137,$$

$$b_1 = \frac{\sum_{i=1}^{10} x_i y_i - 10\bar{x}\bar{y}}{\sum_{i=1}^{10} x_i^2 - 10\bar{x}^2} = 1,26947, \quad b_0 = \bar{y} - b_1\bar{x} = -3073,794.$$

Regresní přímka má tedy tvar

$$y = -3073,794 + 1,26947x. \tag{4.5}$$

Nyní můžeme spočítat koeficient determinace, který vyjadřuje přesnost navržené regresní funkce. Nejlepší aproximací vektoru $\boldsymbol{Y}$ je vektor $\hat{\boldsymbol{Y}} = \boldsymbol{X}\boldsymbol{b}$, tedy reziduální variabilita regresního modelu je

$$S_e = (\boldsymbol{Y} - \hat{\boldsymbol{Y}})^T (\boldsymbol{Y} - \hat{\boldsymbol{Y}}) = (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{b})^T (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{b}) = 16205532,71.$$

Celková variabilita je

$$S_T \sum_{i=1}^{10} y_i^2 - 10\bar{y}^2 = 284863789,8$$

a hledaný koeficient determinace je roven

$$R^2 = 1 - \frac{S_e}{S_T} = 0,94311.$$

Tzn. že $94,3\%$ bodů je vysvětleno navrženým modelem.

Závěr

Cílem této práce bylo vytvoření matematického modelu, který na základě fotografie ruky určí odhad jejího povrchu. Zpracování bylo provedeno v softwaru Matlab a obsahuje velkém množství operací, aby byly ošetřeny různorodé vstupy fotografií. Na začátku praktické části, jsou popsané požadavky, které by vstupní fotografie měla splňovat. Ale i přes dané požadavky se fotografie velmi liší a především se liší objekt ruky.

Z černobílé fotografie ruky jsme odstranili šum pomocí mediánového filtru a následně jsme adaptivním prahováním získali binární obraz. Jelikož výsledkem prahování nebyl ucelený objekt ruky, bylo potřeba obraz dále zpracovat. Použili jsme algoritmus, který vyhledává všechny objekty v obraze. Jelikož může nastat varianta, že jich algoritmus najde víc, je v programu zahrnuto propojování objektů. To jsme provedli pomocí obálky objektů. Ta se totiž dotýká všech objektů tvořících hranici ruky a pomocí ní jsme určili jejich správnou posloupnost. Tím jsme odvodili které objekty mezi sebou propojit. Pak jsme nalezli vnější 8-spojitelnou hranici objektu. Existují případy, kdy se objekty propojí na špatném místě, jelikož podmínkou je místo nejbližších hranic. Proto je v programu zahrnutá část, která tuto variantu ošetřuje a to tím způsobem, že propojíme pixely hranice k sobě blízké vzdáleností a zároveň daleké indexem. To se provádí dokud se hranice razantně nezkrátí. Zároveň jsme tento algoritmus ošetřili maximální vzdáleností, ve které se pixely propojí. Po tomto kroku jsme úspěšně získali hranici ruky.

V dalším kroku jsme určili parametry ruky, potřebné k výpočtu povrchu. K určení těchto parametrů jsme využili křivosti hranice. Křivost jsme vypočítali v každém pixelu hranice a vykreslili jsme ji do grafu. Pak vždy existuje 13 lokálních maxim, které jsme definovali jako body zlomu. Díky nim jsme rozdělili objekt na objekty prstů a objekt dlaně. Dále jsme provedli postupnou erozi. Pomocí ní jsme získali 8-spojitelné křivky středů prstů a v nich hodnoty poloměrů prstů v dané výšce.

Povrch ruky vypočtený modelem S_{model} jsme vypočítali odděleně pro prsty a dlaň. Prsty jsme aproximovali válci, kde jeden válec odpovídá jednomu pixelu středové křivky prstu, má výšku jeden pixel a poloměr je roven hodnotě pixelu z postupné eroze. Pak součtem plášťů těchto válců jsme získali povrch prstů Sp. Povrch dlaně Sd jsme vypočítali jako součet všech objektových pixelů matice dlaně vynásobený dvěma. K tomu jsme provedli jeho korekci nahrazením ostrých hran (malíková hrana a dvě palcové hrany) zaoblenými hranami ve formě půlkružnice v každé délce hrany.

Tímto způsobem jsme získali povrch deseti vybraných rukou. Abychom ověřili správnost navrženého modelu, bylo u těchto rukou provedeno měření jejich reálných povrchů. K tomu bylo potřeba získat 3D modely, které se následně naskenovaly na 3D skeneru. Z programu GOM Inspect byly získány hodnoty povrchů, které považujeme za reálné.

Jelikož hodnoty povrchu S_{model} neodpovídají reálným hodnotám, zkoumali jsme závislost těchto proměnných. Spearmanovým korelačním koeficientem jsme zjistili, že proměnné jsou monotónně závislé na hladině významnosti $\alpha = 0,01$ a hodnota Spearmanova korelačního koeficientu je rovna $r_S = 0,95152$. Na základě toho jsme navrhli regresní model ve tvaru $y = -3073,794 + 1,26947$. Pro ověření vhodnosti modelu, jsme spočítali koeficient determinace R^2, který říká, že $94,3\%$ bodů je vysvětleno navrženým modelem.

Jak již bylo zmíněno v úvodu, jde o navazující práci k bakalářské práci, která se zabývala stejnou problematikou. Jejím cílem bylo navrhnout aproximace geometrickými

útvary a tělesy a nahradit jimi objekt ruky. Vstupy byly definované rozměry změřené ručně a testováno bylo sedm rukou. Spearmanův korelační koeficient uváděl závislost mezi reálnými a vypočítanými povrchy $r_S = 0,92857$. Výsledkem byl regresní model přímky, jehož hodnota koeficientu determinace byla rovna $R^2 = 0,932$.

V porovnání s výsledky z bakalářské práce je nově navržený matematický model přesnější, jelikož prokazuje vyšší závislost na reálných hodnotách povrchu. Tedy podařilo se nám metodu zlepšit z hlediska přesnosti i automatizace.

Literatura

[1] ANDĚL, Jiří. *Základy matematické statistiky.* Vydání třetí. Praha: Matfyzpress, 2011. ISBN 978-80-7378-162-0.

[2] ANDĚL, Jiří. *Statistické metody.* Čtvrté upravené vydání. Praha: Matfyzpress, 2007. ISBN 80-7378-003-8.

[3] ČERMÁK, Libor a Rudolf HLAVIČKA. *Numerické metody.* Vydání třetí. Vysoké učení technické v Brně, Fakulta strojního inženýrství: AKADEMICKÉ NAKLADATELSTVÍ CERM, s.r.o. Brno, 2016, 110 s. ISBN 978-80-214-5437-8.

[4] DOUPOVEC, Miroslav. *Diferenciální geometrie a tenzorový počet.* Brno: VUT v Brně, FSI, Ústav matematiky, 1999. ISBN 80-214-1470-7.

[5] KALUS, René. Breviář vyšší matematiky. Online. In: *Breviář vyšší matematiky.* Ostravská univerzita, 2001, s. 84-91. ISBN 80-7042-819-8. Dostupné z: http://artemis.osu.cz/skripta/kalus1/kap06.pdf. [cit. 2023-10-16].

[6] KARPÍŠEK, Zdeněk. *Matematika IV: Statistika a pravděpodobnost.* 4. přepracované vydání. Brno: AKADEMICKÉ NAKLADATELSTVÍ CERM, 2014. ISBN 978-80-214-4858-2.

[7] KOSOVÁ, Petra. *Numerical Methods of Image Analysis.* Online, Educational text. Brno: Brno University of Technology. [cit. 2023-05-10].

[8] KLÍMA, Miloš, Martin BERNAS, Jiří HOZNAM a Pavel DVOŘÁK. *Zpracování obrazové informace.* Zikova 4, 166 35 Praha 6: Ediční středisko ČVUT, 1996. ISBN 80-01-01436-3.

[9] MAROŠ, Bohumil a Marie MAROŠOVÁ. *Numerické metody I.* Vysoké učení technické v Brně, Fakulta strojního inženýrství: AKADEMICKÉ NAKLADATELSTVÍ CERM, s.r.o. Brno, 2003, 144 s. ISBN 80-214-2388-9.

[10] PRATT, William K. *Digital Image Processing.* Fourth Edition. United States of America: Wiley, 2007. ISBN 978-0-471-76777-0.

[11] REKTORYS, Karel a spolupracovníci. *Přehled užité matematiky.* Vydání třetí. Praha: SNTL - Nakladatelství technické literatury, 1973. ISBN Sign: 2-1130.019.

[12] ŠTARHA, Pavel. *Aplikace numerických metod zpracování obrazové informace.* Habilitační práce. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2015.

Seznam použitých zkratek a symbolů

$u \equiv u(x_1, \cdots, x_n)$	skalární pole
x, y, z	kartézské proměnné
∇	Hamiltonův operátor
Δ	Laplaceův operátor
C	křivka
s	přirozený parametr (oblouk)
κ	křivost křivky
φ	aproximace funkce f
$P_n(x)$	polynom stupně n
g_i	bázové funkce
$\boldsymbol{\beta}$	neznámé koeficienty, neznámé regresní koeficienty
$\boldsymbol{G}$	Gramova matice
$\mathcal{F}\{f\}(\xi, \eta)$	Fourierova transformace funkce f
$F(\xi, \eta)$	Fourierovo spektrum
$A(\xi, \eta)$	amplitudové spektrum
$f_1 * f_2$	konvoluce funkcí f_1 a f_2
$\mathcal{D}\{f\}(\xi, \eta)$	diskrétní Fourierova transformace funkce f
$F_D(\xi, \eta)$	diskrétní Fourierovo spektrum
X, Y	náhodné proměnné
$E(X), \mu$	střední hodnota
$D(X), \sigma^2, var(X)$	rozptyl
$S(X)$	směrodatná odchylka
$C(X, Y), \sigma_{XY}$	kovariance
$\rho(X, Y)$	korelace
$\bar{X}$	výběrový průměr
S_X^2	výběrové rozptyl
S_X	výběrová směrodatná odchylka

S_{XY}	výběrový koeficient kovariance
r	výběrový koeficient korelace
t, τ	prahová hodnota, testovací kritérium, parametr
$\bar{W}_\alpha$	doplněk kritického oboru
R_i, Q_i	pořadí hodnot náhodné proměnné X a Y
r_S	Spearmanův korelační koeficient
$r_S(\alpha), r_S^*$	kritická hodnota Spearmanova korelačního koeficientu
e	chyba regresního modelu
S^*	reziduální součet čtverců
$\boldsymbol{b}$	bodové odhady koeficientů beta
s^2	bodový odhad rozptylu
Se, S^*_{min}	reziduální variabilita regresního modelu
S_T	celková variabilita
R^2	koeficient determinace
$\mathbf{I}$	obrazová matice
$\mathcal{P}, \mathcal{Q}, p, p_1, p_2$	pixel
(r, s)	pozice pixelu v matici obrazu
$\mathcal{N}_4^P(r, s)$	čtyřsousednost pixelu P
$\mathcal{N}_8^P(r, s)$	osmisousednost pixelu P
$\mathcal{N}_D^P(r, s)$	diagonální sousednost pixelu P
R, G, B	intenzity složek barvy
w	dinamický rozsah obrazu
T	atributová množina
$\mathcal{O}$	objekt
$\partial\mathcal{O}$	množina všech hraničních pixelů $\mathcal{O}$
Γ	digitální křivka
γ	po částech lineární křivka
A	ideální obraz bez šumu

S	obraz šumu
B	digitální obraz
$\boldsymbol{H}$	konvoluční jádro
$\boldsymbol{H}_I$	filtr typu identický obraz
$\boldsymbol{H}_L$	filtr typu dolní propust
$\boldsymbol{H}_H$	filtr typu horní propust
h	histogram
A_D	obraz vytvořený dilatací
A_E	obraz vytvořený erozí
V	vstupní matice obrazu, binární
o.č.p	objektová čísla pixelů, rozlišují objekty
O	objektová matice, obsahující o.č.p
H	matice hranicí objektů
$vektorH$	vektor hranicí objektů
VO	matice obálky objektu ruky
HO	matice hranice obálky
$vektorHO$	uspořádaný vektor hranice obálky
HV	matice 8-spojitelné hranice ruky
$vektorHV$	uspořádaný vektor 8-spojitelné hranice ruky
$o1 - o8$	okolní pixely
OK	matice okolních pixelů
M	matice vzdáleností pixelů ve $vektorHV$, dolní trojúhelníková
vzd	vzdálenost pixelů
$delkaHVS$	proměnná délky hranice ruky
$vektorHVS$	nový uspořádaný vektor 8-spojitelné hranice ruky
HVS	matice nové 8-spojitelné hranice ruky
VYS	matice výsledného objektu ruky
K	vektor křivostí hranice ve vektoru $vektorHVS$

ER	matice obsahující hodnoty postupné eroze
SE	matice obsahující středové křivky prstů, SEp k ní pomocná
maxVZD	maximálně možná vzdálenost pixelů jednoho prstu
$vzdVys$	vektor výsledných středů prstů
$bodK$	kontrolní bod ležící na hranici středu zápěstí
HD	matice hranicí rozdělených objektů prstů a dlaně
D	matice objektu dlaně
bh	uspořádaný vektor hranice dlaně
$bodyZ$	body zlomu
tz	tloušťka zápěstí
sz	šířka zápěstí
Sp	povrch prstů
Sd	povrch dlaně
pm	průměr malíčku
pu	průměr ukazováčku
pp	průměr palce
vm	délka malíkové hrany
$vp1$	délka palcové hrany nad palcem
$vp2$	délka palcové hrany pod palcem
S_{model}	povrch navrženého matematického modelu

Seznam příloh

1. SPUSTIT.m - program pro spuštění celého výpočtu povrchu ruky
2. a_Prahovani_adaptivne.m - program, který ze vstupní fotografie vytvoří prahovaný obraz
3. b_Uprava_po_prahovani.m - program, který z prahovaného obrazu získá 8-spojitelnou hranici ruky
4. c_Krivost_hranice - program, který pomocí 8-spojitelné hranice ruky najde body zlomu
5. d_Povrch_ruky - program, který na základě bodů zlomu rozdělí objekt ruky a vypočítá povrch modelu
6. Fotografie - složka obsahující testované fotografie rukou