

bakalářská práce

Ovládání robotu sledujícího člověka gesty

Kovbasjuková Oxana



Červen 2018

Ing. Jan Chudoba

České vysoké učení technické v Praze
Fakulta elektrotechnická, Katedra řídicí techniky

Poděkování

Děkuji vedoucímu za cenné rady, trpělivost a vstřícnost. Rodičům za pomoc při studiu a okolí za všechnu poskytovanou podporu. Také jsem ráda za kolektiv ve škole, ve kterém se rozvíjí diskuze o otázce života, vesmíru a vůbec.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Abstrakt

Tato práce řeší detekci ruky pomocí kamery Kinect a rozpoznávání gest. Cílem této práce je vytvořit podklad pro řízení robota, který bude interagovat s člověkem. Zaměřila jsem se na charakteristické rysy ruky a rozpoložení prstů. Práce představuje algoritmus, který využívá pouze euklidovské vzdálenosti. Data z kamery jsem zpracovávala pomocí openFrameworks s využitím doplňku ofxKinectV2.

Klíčová slova

kamera Kinect; hloubkový obraz; detekce ruky; rozpoznávání gest; HRI; počítačové vidění

Abstrakt

This thesis deals with hand detection and gesture recognition from video stream captured by Kinect camera. The aim of thesis is to create data to control robot for human-robot interaction. Focus was on features of the hand and fingers. Thesis presents an algorithm based on euclidean distances. Data was processed using openFrameworks software with ofxKinectV2 addon.

Keywords

Kinect camera; depth video; hand detection; gesture recognition; HRI; computer vision

Obsah

1	Úvod	1
2	Stávající metody a aplikace	3
2.1	Stávající metody zpracování obrazu	3
2.1.1	Rozostření	3
2.1.2	Prahování	4
2.1.3	Nalezení kostry	4
2.1.4	Detekce hran	6
2.2	Rozpoznávání v obraze	6
2.2.1	Rozpoznávání postav	6
2.2.2	Rozpoznávání částí těla	9
2.2.3	Rozpoznávání gest	9
	Rozpoznávání prstů	9
2.3	Aktuální aplikace rozpoznávání pomocí kamery	11
2.3.1	Možnosti využití gest při ovládání mobilního robotu	11
2.3.2	Konkrétní aplikace používající Kinect nebo rozpoznávání gesta .	11
	Rehabilitace	11
	Využití ve zdravotnictví	12
	Sport	12
	Ovladač Bixi	12
	DJI drony	12
3	Implementace rozpoznávání gest	13
3.1	Software	13
3.2	Hardware	13
3.2.1	Výstupy z kamery	13
	RGB video	14
	Hlubkové video	14
3.3	Postup	14
	Úvodní nastavení	14
	Nalezení ruky	14
	Identifikace gesta	15
	Vykreslení ruky	16
3.4	Předzpracování obrazu	16
3.4.1	Filtrace	16
3.4.2	Prahování	17
3.5	Detekce ruky	18
3.6	Detekce dlaně	19
3.7	Oblast zájmu (ROI)	20
3.8	Detekce prstů	20
3.8.1	Detekce směru prstů	20
3.8.2	Detekce konců prstů	21
	Nejvýše položený bod	21
	Nejvzdálenější bod	21
	Nejvzdálenější bod dle pozice prstu	22
	Nejvzdálenější bod z povoleného okolí	22
3.8.3	Identifikace prstů	22

3.9 Vyhodnocení výsledků detekce gest	22
4 Závěr	23
Literatura	24

Zkratky

Seznam používaných zkratk v bakalářské práci.

HRI	"human-robot interaction"...interakce člověka a robota
OF	openFrameworks ... nástroj pro kreativní programování
FBO	"frame buffer object"...objekt pro vykreslování v OF
HMM	"Hidden Markov Model"...Skrytý Markovův model
P2DHMM	"Pseudo-2D Hidden Markov Model"
VR	virtuální realita
DOF	"degree of freedom"...stupeň volnosti
SVM	"support vector machines"...SVM klasifikátor s podpůrnými vektory
TOF	"time of flight"...doba letu aneb hloubkové kamery založené na výpočtu vzdálenosti z doby letu laseru
ROI	"region of interest"...oblast zájmu
RBF	"radial basis functions"...radiální báze funkce
RGB	"red green blue"...barevný model obsahující složky červená, zelená
1D	jednodimenzionální
2D	dvoudimenzionální

1 Úvod

V dnešní době robotizace, kdy lidé přicházejí na výhody využívání inteligentních zařízení, je snaha rozšířit jejich využití co nejvíce. Ovládání gesty značně pomůže intuitivnímu řízení robotů a usnadní jejich využití. Z pohledu spolupráce běžného uživatele s robotem se řadí řízení zařízení za pomoci kamery mezi nejjednodušší způsoby, jelikož kamery jsou relativně levné a dostupné. Není k tomu potřeba nic jiného, než co většina populace (alespoň cílové skupiny - lidí používající počítače) už má. Jedná se o jednodušší variantu i s ohledem na implementační čas.

Mnoho podobných aplikací již existuje, nebo se zaměřuje na jednotlivé podčásti, často ale vyžadují složitou instalaci a konfiguraci programů. Velké množství pak ani nepodporuje zpracování s malou časovou odezvou a jsou výpočetně složité a časově náročné. Jelikož má být výsledný algoritmus použitelný pro robota, který bude následovat člověka, musí splňovat několik požadavků. Jedná se o systém reálného času a tam musí být odezva dostatečně malá s ohledem na člověka. Zvolená gesta musí být intuitivní a snadno proveditelná.

Následně zvolené přístupy v této práci jsou psány tak, aby měly co nejmenší odezvu. V aktuálních aplikacích se obvykle využívají strojově naučené algoritmy. Pro snazší dosažení rychlejšího zpracování obrazu, které vyhovuje požadavkům, lze využít vlastností obrazu, jako jsou vzdálenosti a úhly k rozpoznávání ruky.

2 Stávající metody a aplikace

Kapitola pojednává o aktuálně používaných metodách v rozpoznávání v obraze. Jsou zde rozoberány aplikace, které využívání kameru Kinect, i potenciální použití v dalších oborech.

2.1 Stávající metody zpracování obrazu

Cílem kapitoly je rešerše aktuálních metod rozpoznávání lidí, těla a jeho částí. Nezbytnou součástí je i tomu předcházející úprava načteného obrazu. Berou se zde v potaz stávající metody, které mohou sloužit k účelům této práce.

Předzpracování obrazu se týká všech aplikací využívajících digitální obrazy z kamery. Při zpracování digitálních obrazů je potřeba obraz rozostřit kvůli odstranění extrémů z obrazu, následně jsou probírány již věci specificky pro zpracování obrazu k detekci objektů.

2.1.1 Rozostření

Rozostření slouží k eliminaci šumu v obraze. Nejosvědčenější metody jsou nízkofrekvenční filtry jako je Gauss a medián. Tato kapitola čerpá ze závěrečné práce Pikory [1], přesněji kapitoly 5 a 6.

Filtr medián se zaměřuje na vyhlazování lokálních extrémů v obraze. Nezachovává jemné čáry a ostré rohy, takže je vhodný pouze v aplikacích, kde na tom nezáleží. Výraznou výhodou představuje v odstranění extrémů, které v obraze mohou vzniknout. Narozdíl od průměrování obraz není zkreslený kvůli chybným pixelům, které mají výrazně odlišnou hodnotu.

Velikost okolí zahrnutého v analýze lze zvětšit při větší hustotě šumu. Pokud v aplikacích záleží na hranách, je důležité zachovat určité rozlišení. Na obrázku 1 lze pozorovat rozdíly mezi filtrem s průměrováním (b) a mediánem (c).



Obrázek 1 a) původní obraz b) průměrování c) medián
Převzato z [1]

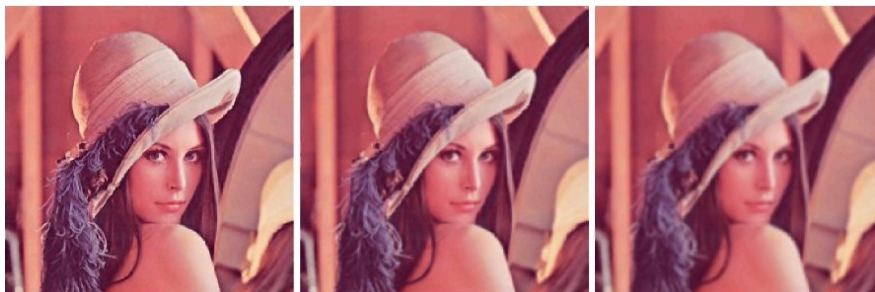
Gaussovský filtr je průměrování s Gaussovským rozložením. Využívá se k vyhlazování obrazu, odstranění detailů a šumu. Obrázek 2 zachycuje rozdíl mezi opakovanou aplikací Gaussovského filtru. Výpočet hodnoty pro každý pixel je uveden ve vzorci:

$$G(x, y) = \frac{1}{2\pi\sigma^2} * e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (1)$$

kde x a y jsou souřadnice daného pixelu a σ je směrodatná odchylka.

V konvoluční masce se přidělí větší váha bodu ve středu, aby se zabránilo rozšíření rozostření do nekonečna. Aby se aplikací konvoluční masky neměnila světlost, součet všech složek konvoluční matice dává hodnotu 1. Příklad konvoluční masky:

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (2)$$



Obrázek 2 a) původní obraz b) jednou aplikovaný Gaussův filtr c) třikrát aplikovaný Gaussovský filtr Převzato z [1]

2.1.2 Prahování

Většina metod se zakládá na zpracování binárního obrazu. Princip je ve zvolení prahové hodnoty, se kterou se všechny hodnoty porovnají. Pokud hodnota přidělená (v případě RGB kamery) nebo načtená (v případě hloubkové kamery) větší než prahová, do výsledného obrazu se uloží '1', v opačném případě se zapíše '0'. Objekt, který je středem zájmu, se skládá z pixelů s hodnotou 1 zatímco ostatní mají hodnotu 0. Metoda prahování, anglicky "thresholding" se dělí na více druhů: lokální, pásmové, polo-prahování, multi-prahování a další. [2]

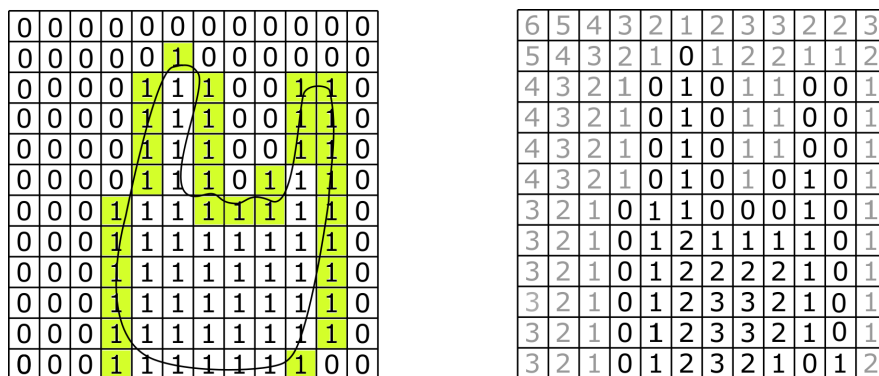
2.1.3 Nalezení kostry

Při hledání topologie a tvaru objektu lze použít například skeletonizaci. Ve zjednodušené představě se jedná o zredukovaný objekt, který představuje minimální dostačující reprezentaci zkoumaného objektu, který obsahuje všechny významné body. Tato kapitola vychází z Palagyoiova přehledu [3]. Existují tři způsoby hledání kostry :

- detekce ze vzdálenostní mapy,
- výpočet Voronoiova diagramu, který je generován pomocí hraničních bodů a
- ztenčení.

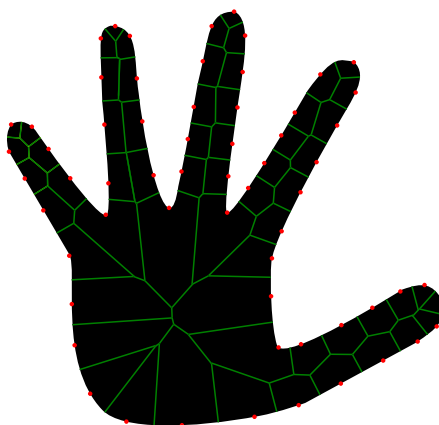
Jelikož nelze zaručit, že se jedná o přesnou kostru objektu, jsou zde dva rozhodovací faktory. Jedná se o požadavek zachování topologie a geometrickou podmínku, která vynucuje umístění kostry uprostřed objektu. Topologickou podmínku splňuje Voronojův diagram a ztenčování, geometrickou zaručuje detekce ze vzdálenostní mapy a Voronojova kostra.

Detekce ze vzdálenostní mapy lze popsat třemi kroky. Nejdříve se v binárním obraze označí hraniční body objektu (zelené pixely na obrázku 3). Poté se vygeneruje vzdálenostní mapa podle toho, jak daleko se nachází konkrétní pixel od nejbližšího hraničního bodu. Hrany kostry reprezentují pixely, které jsou lokálními extrémami, ve výsledné vzdálenostní mapě.



Obrázek 3 Zelené pixely na levém obrázku označují hranice zpracovávaného objektu. Vůči nim se vypočítává vzdálenost zbylých pixelů, z nichž lokální maxima představují kostru. Výpočet probíhá pomocí čtyř sousedních pixelů. Vlastní zpracování.

Voronoiovy diagramy dělí prostor na části podle množiny bodů tak, aby se v každé oblasti nacházel právě jeden bod. Do množiny bodů se přidá část hraničních bodů objektu (červené body na obrázku 4). Sousedící okraje nalezených částí tvoří kostru. Čím více bodů množina obsahuje, tím přesnější kostra bude.



Obrázek 4 Červené puntíky reprezentují zvolené body na kraji ruky. Zelené čáry zobrazují hranice sousedících oblastí. Vlastní zpracování.

Ztenčování je proces, který se opakuje tak dlouho, dokud se v obraze vyskytují pixely, jejichž odstranění neporušuje předem definované podmínky a tudíž se výsledná kostra stále ztenčuje a zároveň zůstává zachována topologická podmínka. Podmínky se mohou lišit podle toho, k jakým účelům je algoritmus používán.

2.1.4 Detekce hran

K rozpoznávání v obraze lze dobře využít nalezených hran. Za hrany lze považovat místa, ve kterých se výrazně mění jas. K výpočtu hran se používá gradient, což je vektorová veličina určující směr a strmost největšího růstu funkce [2].

Mezi vysokofrekvenční filtry, které se používají na zpracování digitálního obrazu, patří i Laplaceův filtr. Ve výpočtu se užívá gradient. Laplaceův filtr využívá pouze velikost gradientu a pro její odhad se používá všesměrový operátor vycházející z parciálních derivací. Nevýhodou je, že některé hrany detekuje dvakrát. Vzorec pro výpočet hodnoty pixelu je:

$$\nabla^2 g(x, y) = \frac{\delta^2 g(x, y)}{\delta x^2} + \frac{\delta^2 g(x, y)}{\delta y^2} \quad (3)$$

Laplacián je aproximován diskretní konvolucí. Příklad konvolučního jádra může být:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4)$$

2.2 Rozpoznávání v obraze

2.2.1 Rozpoznávání postav

Mezi hlavní možnosti přístupů k implementaci detekce postav z obrazu patří:

- na základě modelu,
- na základě prvků (vlastností),
- strojově naučené programy,
- neuronové sítě, nebo
- transformační matice vzdálenostní mapy [4].

Jednou z metod je dle Ch. Nakajima a spol. [5] ukládání několika obrázků, z nichž se detekuje pohybující se objekt a vše, co se nehýbe, se označí jako pozadí a ignoruje. Následně se naleznou kraje možné postavy, aby se vylepšil výsledek a vymazal šum. Pokud se jedinec pohybuje v druhém kroce, oříznutí probíhá z posledního získaného obrazu. Pokud nebyl detekován žádný jedinec, tak se do paměti uloží pozadí, které se z následujících obrazů rovnou vymaže pro větší přesnost. K vlastní identifikaci se používá SVM (support vector machines) a k-NN klasifikátor.

Hussein a spol. [6] rozpoznávají postavy tím, že prohledají obraz a porovnají siluety s databází, ve které jsou vzory lidských postav ve formě binárního obrazu. Shoda je detekována pomocí zkosením objektů. Vzdálenost D mezi vzorem V a objektem O z obrazu se počítá vzorcem

$$D(O, V) = \frac{1}{|V|} \sum_i C_i V_i, \quad (5)$$

kde $|V|$ je počet pixelů v siluete ve vzoru, T_i je hodnota pixelu i ve vzoru a C_i je rozdíl zkosení pixelu i v obraze. Čím menší hodnota mezi vzorem a obrazem, tím lepší shoda je detekována.

Nalezení podobnosti zkosením předpokládá již nalezené hrany objektů v obraze. Vzor se překryje s objektem a transformací bodů jednoho objektu pomocí parametrických

transformačních rovnic.

Jiný přístup nalézá objekt v zorném poli pomocí hloubkové kamery. Následuje rozhodnutí, zda objekt má lidské nohy nebo aspoň dva objekty odpovídající tvaru, a pokud ano, tak se přejde k dalšímu kroku, ve kterém se detekuje barva kůže z RGB vstupu. Nalezený výsledek slouží jako oblast k detekci obličeje, čímž je nalezen člověk [7].

Gavrila [8] se snaží snížit výpočetní náročnost tak, že nahrazuje procházení obrazu postupně schopností zaměřit se na jeden úryvek přímo a ve druhém kroce teprve hledá podobnosti tvarů. Dosahuje tak zpracování bez časové prodlevy. Shodu podle tvaru detekuje pomocí vzdálenosti křivek, která se počítá podobně jako u Hussein a spol. [6] podle vzorce:

$$D(O, V) = \frac{1}{|V|} \sum_{t \in T} d_I(t), \quad (6)$$

kde $|V|$ je počet prvků ve vzoru a $d_I(t)$ představuje vzdálenost daného prvku ve vzoru a odpovídajícímu v obraze. Pokud výsledná hodnota je menší než předem určená prahová hodnota, považuje se objekt v obraze za postavu. Detekce chodce podle vzoru lze pozorovat na obrázku 5.



(a)



(b)



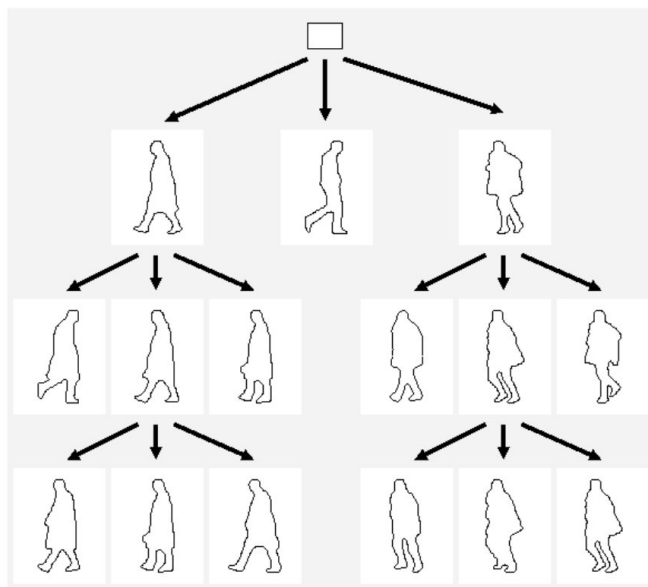
(c)



(d)

Obrázek 5 a) původní obraz b) vzor c) nalezené hrany d) vyobrazené vzdálenosti Převzato [8]

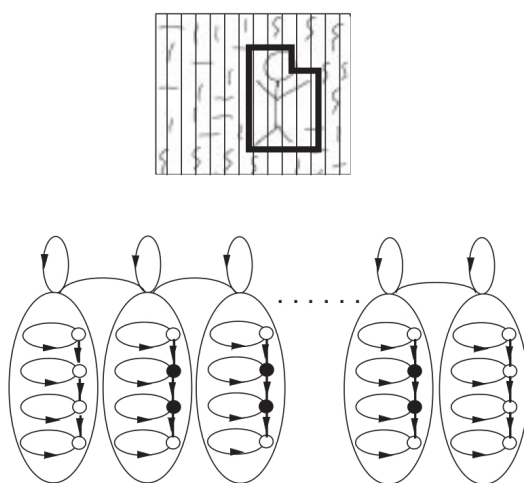
V rámci optimalizace se tato metoda rozšiřuje na hierarchizovanou, která sdružuje podobné vzory do shluků a tudíž se prohledávání databáze zrychlí (viz obrázek 6).



Obrázek 6 Využití hierarchie v databázi vzorů. Převzato z [8]

Ověření výsledku probíhá pomocí RBF klasifikátorem (neuronová síť založená na radiálních básových funkcích). Nejdříve se z původního obrazu vybere čtyřúhelník obsahující možnou postavu a na základě euklidovské vzdálenosti určuje, jestli jednotlivé pixely jsou součástí postavy či nikoliv.

Riggol a spol. [9] kombinují rozpoznávací metody založené na prvcích a na modelech za cílem dosažení spolehlivého a efektivního programu. Využívají P2DHMM (Pseudo-2D Hidden Markov Model). Jedná se o předem naučený algoritmus, který dokáže na základě pravděpodobností určit, jestli se jedná o postavu nebo ne. Příklad detekované postavy a postupné rozhodování je vidět na obrázku 7.



Obrázek 7 Stochastický model 2D objektu. Převzato z [9]

2.2.2 Rozpoznávání částí těla

Ruka po zápěstí má 23 stupňů volnosti a v kombinaci s různým osvětlením a změnami v pozadí se jedná o značně komplikovaný problém na řešení pomocí podobnosti.

Jestliže aplikace nevyžaduje přizpůsobení okolí (pozadí obrazu) a zůstává na stejném místě, lze použít rozdíl mezi objektem a pozadím [10]. Vyžaduje to co nejvíce neměnné pozadí a dostatečný rozdíl mezi objektem a zbytkem. Rozdíl lze hledat jednoduše procházením pole nebo využitím "chytrého" algoritmu, který spočívá v identifikaci části obrazu, která se pohybuje oproti posledním obrazům. Příklad detekce ruky odstraněním pozadí je ukázán na obrázku 8. Pro účely bakalářské práce je metoda nevhodná, jelikož je potřeba řídit robota, který následuje člověka a tudíž nelze spoléhat na stejné okolí.



Obrázek 8 a) vstup b) referenční obraz c) výstup Převzato z [10]

2.2.3 Rozpoznávání gest

Mezi rozpoznávání na základě vzhledu se řadí mnoho metod zahrnujících strojově naučené algoritmy jako neuronové sítě, HMM (Skrytý Markovův model) a jiné [11]. Gesto lze identifikovat i určením pozice a směru ruky a jednotlivých prstů.

Kim a spol. [12] používají bílé značky na špičkách prstů, které sledují černým světlem a detekují jednoznačně špičky prstů, což umožňuje velkou škálu možných gest. Na druhou stranu vzniká i omezení barvy pozadí, které lze eliminovat jen v určitém prostředí, tudíž se jedná o vhodnou variantu k využití při hrách ve VR (virtuální realitě).

Shaker a Zliekha [13] získávají směr prstu ruky, kterou snímají dvě kamery. Jedna kamera natáčí shora zatímco druhá z boku. Gesto se rozpoznává z každého vstupu zvlášť. Nejdříve přepočítají obraz na stupně šedi a poté dle určené prahové hodnoty odfiltruje pozadí a vytvoří binární obraz. V tomto kroce je třeba, aby bylo pozadí jednoduché a jasně rozlišitelné od ruky. Kvůli odstranění šumu se obrázek rozmaže a následně zaostří. Je důležité, aby byla ruka největší objekt v obraze. Poté se pomocí Laplaciánu získají hrany daného objektu a z nich se vytvoří kontury. Gesto se identifikuje nalezením největšího vrcholu v obraze a nalezením hrany, kterou opisuje natažený prst a následnou kombinací obou směrů do 3D. Postup je znázorněn na obrázku 9.

Rozpoznávání prstů

Hackenberg a spol. [13] implementovali identifikaci prstů a dlaně na základě několika kroků. Nejprve projde hloubkový obraz z kamery a hledá tvary podobné špičkám prstů nebo rourovitěho tvaru podobného prstům. Poté se zaměří na detailnější vlastnosti



Obrázek 9 Popis algoritmu dle Shakera a Zliekha. Přeloženo a převzato z [13].

dlaně. Z nalezených vhodných polí, která mohou představovat dlaně se vyřadí ta, která nejsou napojena na prsty, nebo jsou menší než předem určená hodnota (odvozená od průměrně velikosti hlavy) a vzdáleností špiček prstů od dlaně.

Bez potřeby porovnávání tvarů a zdrojů se dají rozpoznat jednotlivé špičky prstů nalezením lokálního maxima v obrysu ruky. Ve všech adekvátních směrech se naleznou kandidáti a následně vyloučí mylné identifikace. Jedná se o postup náchylný na šum v obraze, podobná odolnější metoda navrhuje porovnávání vzdálenosti obrysu k pozici ruky [11].

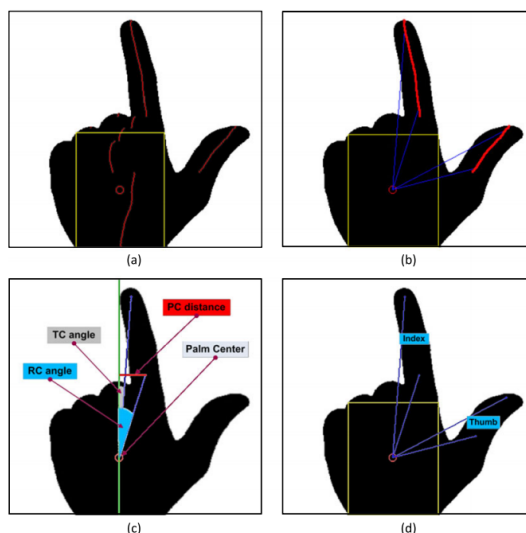
Jednotlivé prsty lze rozpoznat například pomocí vyhledávání na základě tvaru [14]. Jde o zjednodušení představy objektu na geometrický objekt a vyhledávání vhodných kandidátů v obraze, které mají tyčovitý tvar a délku v definovaných mezích.

Mezi další programy rozpoznávající špičky prstů z již relativně čistého čtyřúhelníku, který obsahuje zájmový objekt, patří identifikace na základě dvou specifických vlastností. Střed špičky prstu je obklopen pixely patřící prstu a ty jsou obklopeny dlouhou řadou pixelů nepatřících prstu [10]. Ruce lze identifikovat i po nalezení prstů tím, že se vyloučí objekty podobného tvaru (propisky, fixy), které nesplňují vlastnosti rukou, například nejsou napojeny na žádnou dlaň. Nutná podmínka může být definována jako velikost úhlů mezi prsty. Vzdálenosti pixelů (bodů na ruce) lze počítat dle obrázku 10 pomocí vzorců:

$$RC_{angle} = 90 - \tan^{-1} \frac{y_r - y_{pc}}{x_r - x_{pc}},$$

$$TC_{angle} = 90 - \tan^{-1} \frac{y_{ft} - y_{pc}}{x_{ft} - x_{pc}},$$

kde RC_{angle} je úhel mezi přímkou vedenou vertikálně středem dlaně a spojnici se středem počátku prstu, TC_{angle} je úhel mezi přímkou vedenou vertikálně středem dlaně a spojnici se špičkou prstu, x_{pc}, y_{pc} souřadnice středu dlaně, x_r, y_r souřadnice středu počátku prstu a x_{ft}, y_{ft} souřadnice špičky prstu.



Obrázek 10 Vyobrazení definovaných pojmů, které lze využít k výpočtům pozic, na základě proporčních vlastností prstů. Převzato z [15].

2.3 Aktuální aplikace rozpoznávání pomocí kamery

2.3.1 Možnosti využití gest při ovládání mobilního robotu

Všechny aplikace, které mohou využít výhod [10] z práce:

- Ovládání elektronických zařízení na velmi malém prostoru
- Ovládání elektronických zařízení z větší vzdálenosti
- Snížený počet součástek (nejsou potřeba klávesnice a myš)
- Zjednodušené ovládání elektroniky
- Potenciál zabezpečení proti zneužití (při implementaci ochranných prvků)
- Minimalistické (méně tlačítek a obdobných prvků)

Potenciál pro využití:

- Překlad znakové řeči
- Ovládání chytrých zařízení (chytrá domácnost - světla, hudba, televize, telefon)
- Prezentace před publikem (ovládání počítače)

Již využíváno:

- Rozpoznání chodce v dohledu vozidla
- Sociální experimenty (vyslání robotu stopovat)
- Ovládání a hraní na konzolích

2.3.2 Konkrétní aplikace používající Kinect nebo rozpoznávání gesta

Rehabilitace

Mezi využití kamery a rozpoznávání gest v lékařské oblasti patří převážně rehabilitační cvičení. V nemocnici ve městě Reading využívají kameru Kinect pacienti po mrtvici na zlepšení pohyblivosti a koordinace [16].

Využití ve zdravotnictví

Kinect lze dobře využít jako uživatelské rozhraní během operace, když chce doktor přístup k záznamům pacienta a nemusí se tak dotýkat žádných nesterilních věcí [17].

Sport

Kamera Kinect se využila i v experimentu provedeném na univerzitě v Sarawaku [18]. Cílem experimentu bylo nalezení vztahu mezi tepem a pohyby hráče badmintonu. Kamera snímala hráče při hře a měřila preciznost jednotlivých úderů. Výsledkem bylo, že při zvýšeném tepu klesá podávaný výkon a tudíž lze lépe zamezit omdlávání při přetížení sportovců.

Ovladač Bixi

Bixi je bezdotykový ovladač k mobilnímu zařízení. Hlavním zaměřením je usnadnění ovládání vedlejších aplikací během řízení auta tak, aby se řidič mohl soustředit na provoz. Po spárování se zařízením přes bluetooth lze gesty volit adresu pro navigaci, ztlážit či zastavit hudbu, ztlumit jas obrazovky, přijímat hovory a jiné [19].

DJI drony

Drony od společnosti Da-Jiang Innovations [20] již také využívají rozpoznávání gest k ovládání. Mavic, Phantom 4 Pro, Advance a Spark podporují focení, které se zapne vykreslením žádaného rámečku fotky rukou.

Spark podporuje mnohem více funkcí ovládaných gesty. Po natažení ruky směrem dopředu a nahoru k dronu pod úhlem 45° se začne nahrávat video. Podporuje také vznášení se z dlaně a přistávání na dlaň. Pro vzlet je třeba držet dron na dlaně kamerou k obličeji, aby ho to rozpoznalo, a následně se vznese. Po mávnutí rukou před kamerou jedním ze čtyř hlavních směrů se dron posune daným směrem. Při zamávání začne dron následovat trajektorii chůze člověka. Při detekci dlaně pod sebou na ni dron přistane [21].

3 Implementace rozpoznávání gest

Následné algoritmy byly testovány na kameře Kinect v2 s využitím nástroje projectGenerator od openFrameworks. Program má i vývojové prostředí, ze kterého lze spustit aplikaci. V příloženém souboru bakalářské práce README.txt je uveden postup pro spuštění programu pomocí generátoru v příkazové řádce.

3.1 Software

OpenFrameworks [22] je open source C++ nástroj pro kreativní programování. Využívá doplněk ofxKinectV2 [23]. Oproti zabudovanému ofxKinect je optimalizovaný pro aktuální openFrameworks (verze 0.9.0), je stabilnější, rychlejší a podporuje pro případné potřeby i více kamer.

Kód v openFrameworks se dělí do třech hlavních částí. Jedná se o funkce *setup()*, *update()* a *draw()*. Sekce *setup* slouží pro počáteční nastavení programu, proměnných apod., *update* obsahuje výpočetní a aktualizací část a *draw* má na starosti vykreslování.

Program se snaží vykonávat všechny části tak často, jak to lze. V *update* i *draw* se může využít funkce *ofGetElapsedTimef()*, která vrací vteřiny v desetinné přesnosti od spuštění programu nebo *ofGetElapsedTimeMillis()*, která vrací čas od resetování čítače. Jelikož lze předpokládat, že člověk nemění gesta rychleji, než je počítač zpracovává, tak můžeme využít funkci *modulo* (i jiných) a ovlivnit jak často se budou vykonávat *update* a *draw* nebo jejich podčásti. Funkce *isFrameNew()*, která vrací binární hodnotu určující, jestli se snímek změnil či nikoliv, omezuje počet snímků, které jsou následně zpracovány, aby se neprováděly výpočty opakovaně.

3.2 Hardware

K implementaci této práce byla využita kamera Kinect v2. Zdroje, ze kterých lze data využít k potřebám aplikace jsou RGB kamera s rozlišením 1920 x 1080 pixelů a kamera na snímání hloubky s rozlišením 512 x 424 pixelů.

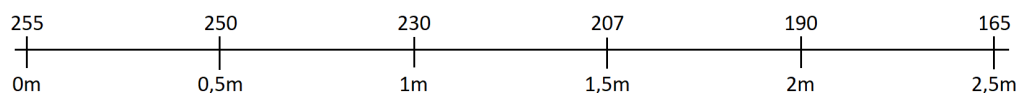
Jedná se o relativně levnou a dostupnou kameru, jejíž cena se v době zpracování této práce (začátek roku 2018) pohybovala do tří tisíc korun.

3.2.1 Výstupy z kamery

Obě kamery poskytují užitečné informace pro účely projektu. RGB kamera nabízí pole pixelů, kde každý má složky RGB s hodnotami v intervalu 0 až 255. Lze využít v kombinaci s robustním programem, který by na základě barvy, tvarů a dalších vylučovacích prvků správně detekoval nejdříve ruku a poté i gesta.

Hloubková kamera poskytuje pole s hodnotami 0 až 255 na základě vzdálenosti objektu od kamery. Stupnice hodnot je klesající, to jest nejbližší objekty mají hodnotu 255. Ve vzdálenosti 3 metry hodnoty klesnou na 150.

Minimální vzdálenost pro správné měření vzdálenosti je 0.5 m a maximální 4.5 m. Při větší vzdálenosti bude Kinect stále ještě detekovat objekty v zorném poli, ale bude



Obrázek 11 Stupnice hodnot, kterých nabývá hloubková kamera v závislosti na vzdálenosti

poskytovat nepřesná data. Velký vliv má i hloubkové rozlišení, které je pro velké vzdálenosti nedostačující. V dálce 5 m od kamery je rozlišení 7 cm.

Měření vlastností prostředí, které lze využít jsou vzdálenost od kamery, vzdálenost mezi jednotlivými pixely, úhly a barvy pixelů. V této bakalářské práci je kladen důraz na vzdálenosti.

RGB video

Výstupem z RGB kamery je dvoudimenzionální pole se třemi hodnotami jednotlivých barevných složek. Zpracování obrazu z RGB videa nebude v této práci využito.

Hloubkové video

Zpracování hodnot z hloubkové kamery je jednodušší, protože se jedná o jednodimenzionální pole s hodnotou, která reprezentuje vzdálenost. Prahování (thresholding) se pak provádí jednoduchým porovnáním hodnot a vytvoření binárního obrazu je rychlejší, než komplikované rozhodovací funkce u barevného obrazu.

Nedostatky vyplývající z využití pouze hloubkové kamery spočívají především v tom, že pokud je v záběru objekt, který má podobnou stavbu a strukturu jako je ruka, tak je mylně detekován a zpracováván.

3.3 Postup

Doporučený postup zpracování by měl sestávat ze čtyř částí:

1. úvodní nastavení,
2. nalezení ruky,
3. identifikace gesta a
4. vykreslení ruky (volitelně).

Úvodní nastavení

Uživateli aplikace je potřeba dát na vědomí podmínky interakce. V praktické části bakalářské práce je nutné, aby byla ruka nejbližší objekt v zorném poli kamery. Alternativou může být počáteční umístění ruky doprostřed kamery, aby mohla dále vycházet z umístění za využívání historie nebo aby se zkrátila doba hledání ruky.

Nalezení ruky

V této části je prostor pro určení podmínek a způsobu vyhledávání ruky. Je vhodné nejdříve vyplnit díry vzniklé šumem v obraze a zahrnout další chyby hardwaru a vlivy okolí. Pokud má být aplikace využívána v náročnějším okolí, musí obsahovat i vyloučení

veškerých nevyhovujících objektů, jako jsou objekty podobného tvaru či cizí ruce. Při využívání pouze vzdáleností a úhlů jednotlivých pixelů se obtížně zahrnují všechny možnosti a způsoby provedení gesta. Jelikož mají lidé různé proporce a tvar rukou, to co je pro jednoho obvyklé rozpoložení ruky, může kazit správnou detekci ruky jiného jedince.

Předpoklad pro následující text je vytvoření binárního obrazu z originálního vstupu prahováním. Chybné pixely vzniklé šumem už by měly být eliminovány. Zároveň platí, že čím více podmínek se uživateli na začátku předloží, tím jednodušší je nalezení ruky.

U správně ukazovaného gesta lze předpokládat stejnou vzdálenost jednotlivých bodů ruky od kamery. Mírné odchylky se dají buď zahrnout tolerancí, který rozdíl v určitém rozmezí bude považovat za ekvivalentní, nebo vyloučit úplně, jelikož se dá předpokládat, že to nebude mít vliv na zpracování, pokud se jedná o krajní body. Aby byl program uživatelsky přívětivý, v programu je zvolena tolerance na zachycení těchto rozdílů v načítaných hodnotách.

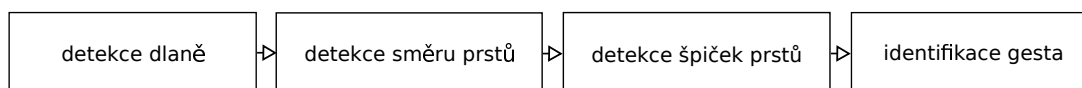
V bakalářské práci se vychází z předem definované vzdálenosti, ve které je nalezen největší možný čtverec, který by představoval dlaň. Vyhledává se největší čtverec v binárním obraze, jelikož z ruky představuje jednoznačně dlaň. Pro účely nalezení musí existovat pole pixelů, které by reprezentovalo dlaň, či případně podobný objekt, který by se následně vyloučil podle dalších kritérií. Nadále je dlaň definovaná středem nalezeného čtverce a jeho šířkou.

Následuje vylučování mylně detekovaných objektů podle toho, jak robustní program má být a v jak obtížném prostředí bude detekce probíhat. V této bakalářské práci se na vylučování neklade velký důraz, jelikož se jedná o komplexní problematiku. Nalezení ruky je rozdělené na jednotlivé úkoly a proto se modul vylučování může snadno upravovat a rozšiřovat v závislosti na složitosti konkrétní skutečné aplikace.

V rámci bakalářské práce je ruka definována tím, že obsahuje prsty, které mají proporcionálně ke dlani určitou velikost a vzdálenost od středu dlaně. Jelikož známe střed nalezené dlaně, lze identifikovat i prsty, respektive konečky prstů. Na počátku programu musí být ruka otevřená se všemi prsty, aby se dala nalézt. Pro ilustraci bude rozebráno gesto se všemi prsty nataženými.

Nejjednodušší metodou je nalezení lokálních maxim po šíři dlaně. Tyto metody jsou více rozoberány v kapitole detekce prstů. Když se dlaň rozdělí na čtyři části a v každé se nalezení lokálních extrémů po ose y (ose x pro horizontální polohu ruky), pro každý prst se změří vzdálenost špičky prstu od středu dlaně. Pokud se bude výrazně odchylovat od proporcí běžné ruky, bude nalezený objekt vyloučen.

Sofistikovanější programy využívají tvorbu skeletonu pro sledování a identifikaci jak postavy, tak i ruky. Jedná se o postup nalezení kostí a jejich kloubů, které jsou reprezentovány čarami, což pak reprezentuje ruku (případně tělo). Jedná se o možný, ale náročnější a zdlouhavější postup.



Obrázek 12 Postup nalezení ruky a identifikace gesta

Identifikace gesta

Gesta lze rozlišovat na statická a dynamická. Pro dynamická gesta je třeba udržovat si v paměti minulé stavy a stanovit dobu, po kterou by bylo přijatelné pro uživatele

provádět gesto a zároveň nemohlo být mylně identifikované. Nejjednodušší je identifikovat gesta statická, kde s využitím euklidovských vzdáleností lze vypočítat počet prstů, a tak v základní verzi bude prostor pro minimálně pět variant, pokud se pro zjednodušení vyloučí zavřená pěst. Program s implementovanou identifikací jednotlivých prstů poskytuje více variant, než by bylo potřeba, nebo by bylo zapamatovatelné pro uživatele.

Počet prstů lze spočítat po správné identifikaci jednotlivých prstů, a jedná se o nejjednodušší způsob klasifikace gesta. Pro přesnost se lze opřít i o úhly a vzdálenosti mezi středem dlaně a špičkami prstů podle vzorců uvedených v kapitole 2.1.4 u obrázku 10.

Sledováním polohy dlaně v následných snímcích lze detekování gesta zefektivnit. Za předpokladu, že se člověk pohybuje pomaleji, než se střídají snímky k analýze, lze procházet menší část pole, například jen okolí místa, kde se posledně nacházela ruka s prsty a kontrolovat změny, zda počet ukázaných prstů se zmenšil nebo zvětšil. K tomuto účelu by stále stačilo gesto definované počtem prstů bez závislosti na tom, o které konkrétně se jedná.

Otázkou je i jak je definováno gesto. Jedná-li se o číslo, které představuje počet vztyčených prstů, je to nejjednodušší. Může to být ale i objekt, který obsahuje pět prstů, z nichž každý může být zvednutý nebo nikoliv. Pak je podstatně přehlednější implementace více než pěti gest. Identifikace je ale náročnější, jelikož každý nalezený prst se musí nejdříve identifikovat, poté aktualizovat objekt představující ruku před kamerou a porovnat s implementovanými gesty.

Vykreslení ruky

Objekty se přes video vykreslují pomocí FBO ("frame buffer object"). Jedná se o buffery s objekty, které je třeba vykreslit. Reprezentují plátno, na které se příkazy *begin()* a *end()* vykreslují 3D objekty a jednou za snímek či méně často (podle požadavků) se vykreslí příkazem *draw()*. Pro lepší přehled se mohou jednotlivé prsty (konečky prstů) zvýraznit koulí, zatímco celá ruka krychlí. Vykreslují se pouze základní objekty, které upřesňují nalezenou pozici rukou a prstů, aby nebylo potřeba udržovat v paměti přesné okraje ruky, které nejsou pro detekci gesta potřeba.

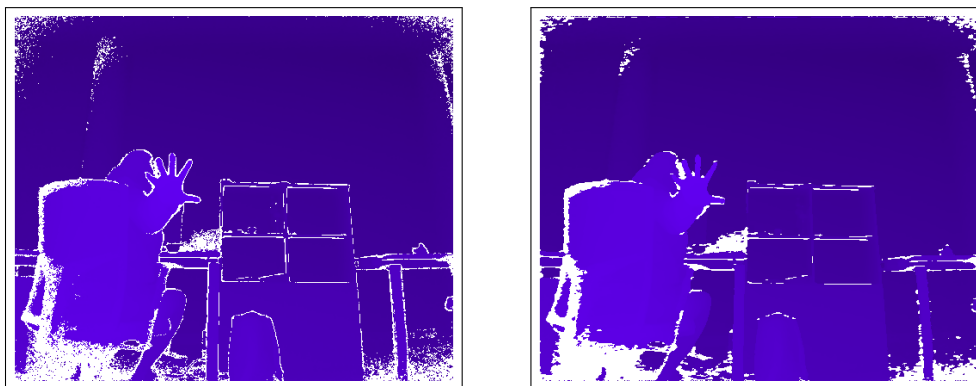
3.4 Předzpracování obrazu

Kvůli přehlednější práci s matematickými parametry obrazu se nejdříve načtou data z kamery, která jsou uložena v jednodimenzionálním poli a transformují se do dvoudimenzionálního pole pro další zpracování. Přepis je potřeba kvůli práci se souřadnicemi.

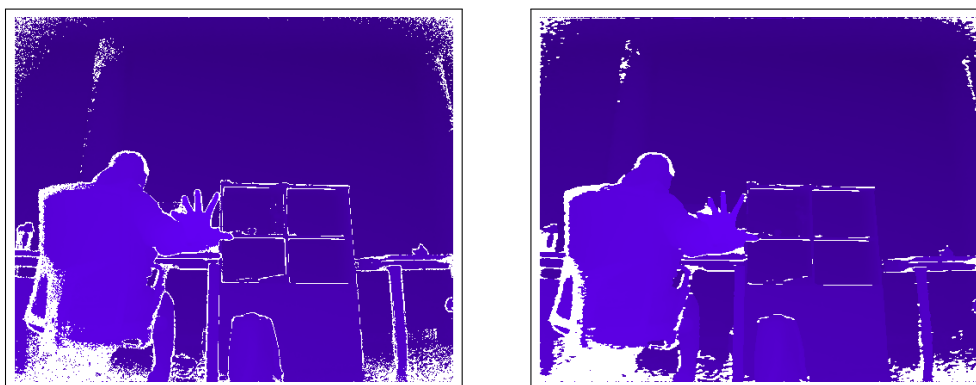
3.4.1 Filtrace

Chybné hodnoty načtené z kamery je potřeba nejdříve co nejvíce eliminovat. Za tímto účelem je použit medián z okolí.

Velikost okolí je zvolena empiricky. Čím větší okolí, tím více je výsledek rozmazaný, ale zato obsahuje menší počet odchýlených hodnot, které by mohly narušit průběh zpracování. Pokud se vezme okolí malé, zůstane jich více, ale lépe se zachovávají tvary. Na následujících obrázcích lze pozorovat rozdíl mezi okolím 9 (obrázek 13) a okolím 25 (obrázek 14).



Obrázek 13 Medián vypočítaný z okolí mohutnosti 9
a) originální obraz b) po úpravě

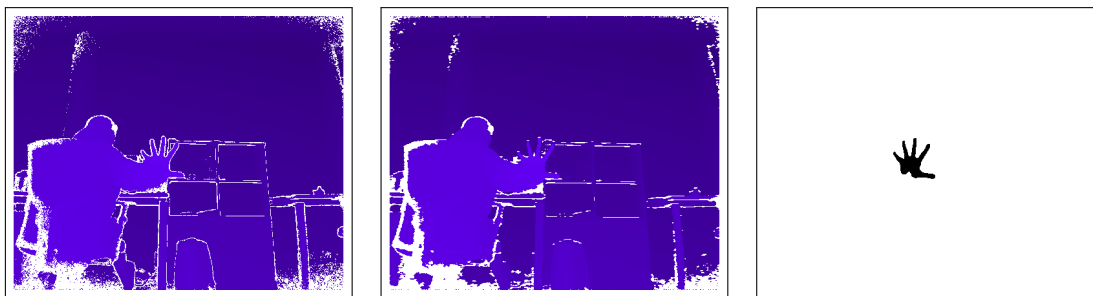


Obrázek 14 Medián vypočítaný z okolí mohutnosti 25
a) originální obraz b) po úpravě

3.4.2 Prahování

Pro manipulaci s nalezenými tvary a výpočty mezi jednotlivými body je snazší pracovat s binárním obrazem. Nejdříve se obraz projde a najde nejbližší objekt (maximální hodnota). Díky předchozí filtraci nezkreslí tuto hodnotu žádný špatně detekovaný pixel, který by měl větší hodnotu. Následně se již porovná každý jednotlivý pixel s hodnotou a vytvoří se pole s hodnotami 1 a 0.

Jelikož lze předpokládat, že člověk nebude mít vždy ruku striktně kolmo k pohledu kamery, je záhodno odečíst od hodnoty představující vzdálenost ruky toleranci. Výsledná hodnota tedy záleží na tom, zda původní pixel byl blíže, než nejbližší objekt s ohledem na odchylku. Tolerance není závislá na vzdálenosti ruky od kamery. Na obrázcích 15, 16 a 17 lze pozorovat změny s ohledem na různé tolerance naklonění ruky. S tolerancí 10 je vidět, že kus dlaně chybí, při toleranci 15 je s rukou načtený i znatelný kus předloktí a k tomu i nejednotný kus jiného objektu. Při toleranci pouze 12 je stále detekováno mnoho pixelů a tak je zvolena 10 jako nejlepší tolerance.



Obrázek 15 Prahování s odchylkou 10
a) originální obraz b) po filtraci c) binární obraz



Obrázek 16 Prahování s odchylkou 12
a) originální obraz b) po filtraci c) binární obraz

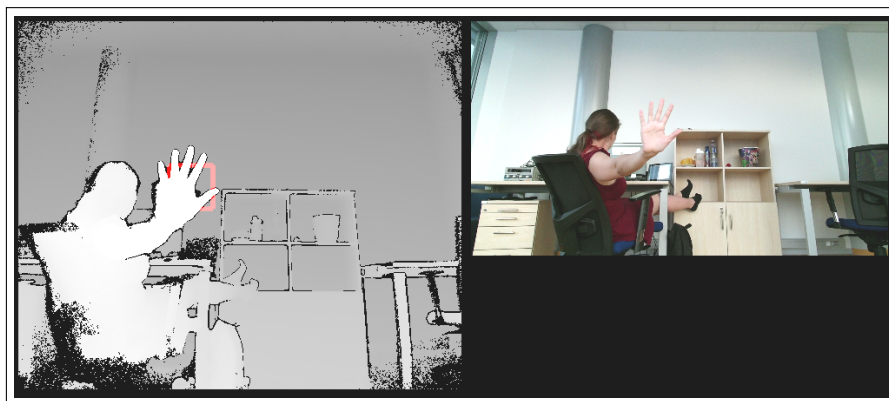


Obrázek 17 Prahování s odchylkou 15
a) originální obraz b) po filtraci c) binární obraz

3.5 Detekce ruky

Definujeme-li ruku jako nejbližší objekt, pak je nalezena již při prahování. Pro lepší názornost se ukládá umístění objektu pomocí pole souřadnic pixelů. Pole obsahuje nalezené pixelů, které patří objektu. V programu je vizualizován červeným čtvercem, který má velikost odvozenou od velikosti pole a střed v souřadnicích X_{avg}, Y_{avg} , které jsou vypočteny jako průměr ze všech souřadnic pixelů, které patří objektu. Na následujících obrázcích je vidět rozdíl mezi vyhledáváním v originálních datech a nebere v potaz chybné hodnoty a vyhledáváním v binárním obraze s vyfiltrovaným šumem.

Z obrázků 18 a 19 lze poznat, že vyhledávání ve filtrovaných datech je přesnější.



Obrázek 18 Obrázek ukazuje největší nalezený nejbližší objekt z originálních dat.



Obrázek 19 Největší nejbližší objekt nalezený v binárních vyfiltrovaných datech.

3.6 Detekce dlaně

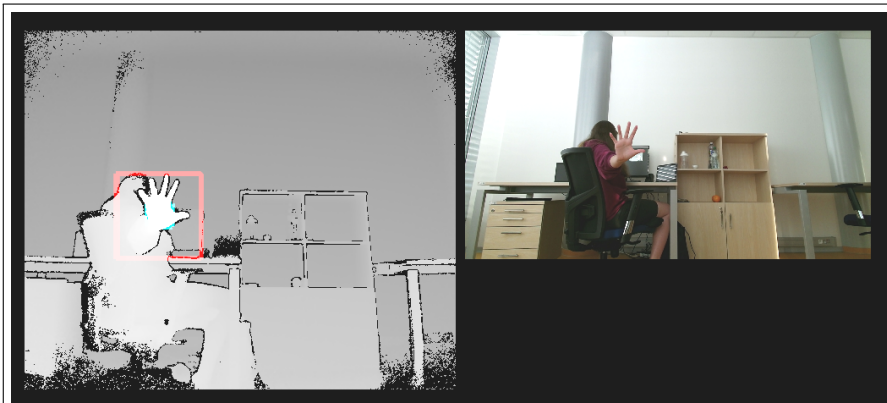
V binárním obraze se vyhledá největší plný čtverec, který lze považovat za dlaň. Vytvoří se pomocná matice, do které se zapíše krajové hodnoty binárního obrazu a poté se binární pole prochází z levého horního rohu. Pokud je na dané pozici binárního obrazu nula, přepíše se i do matice. Pokud je tam ale jedna, pak se nalezne minimum třech sousedů z levého horního rohu. To znamená minimum z horního, levého a šikmo horního nalevo. K minumu se přičte jednička, čímž se zvětší velikost čtvercové matice. Ve výsledku tak největší číslo, které je nalezené v pomocné matici, představuje pravý dolní roh největší nalezené matice a zároveň její velikost [24]. Pro lepší práci v budoucnu je dlaň uložena jako středový bod a velikost matice.

0	1	0	1
0	1	1	1
1	1	1	1
0	1	1	1

0	1	0	1
0	1	1	1
1	1	2	2
0	1	2	3

Obrázek 20 Nalezení největší jednotkové matice v binárním obraze. Na levé mapě je binárního obrazu, na kterém se hledá největší plný čtverec (znázorněn zeleně). Vpravo na pomocné matici je detekován pravý dolní konec největšího čtverce (znázorněn oranžově) a jeho velikost je výsledná hodnota (velikost tři pixely).

Detekce dlaně je vyznačena zeleným čtvercem, jenž má velikost nalezené jednotkové matice a jeho střed má souřadnice X_c, Y_c , které budou následně využívány jako střed dlaně.



Obrázek 21 Červený čtverec znázorňuje polohu a velikost nalezeného nejbližšího objektu. Zelený čtverec obkresluje největší nalezený čtverec v objektu.

3.7 Oblast zájmu (ROI)

Po nalezení předpokládaného umístění dlaně se celá oblast označí jako oblast zájmu ('region of interest'). Tato oblast je přepokopována do vlastního pole nad kterým pak probíhají další algoritmy. Tímto přepokopováním se nepatrně zrychlí běh programu (větší předvídatelnost přednačítání mezipaměti, méně argumentů ve volání funkcí), ale hlavní výhoda spočívá v jednodušší údržbě navazujících algoritmů.

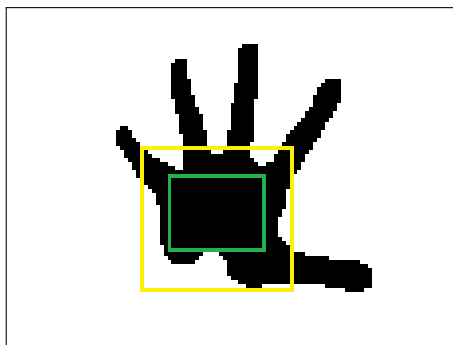
3.8 Detekce prstů

V aplikaci je implementováno několik způsobů nalezení prstů. Pouze některé jsou efektivní a použity ve finální verzi programu, ale pro různost postupu jsou zde uvedeny. Pro všechny byly použity předchozí úpravy obrazu a detekce směru prstů.

3.8.1 Detekce směru prstů

Za předpokladu, že jsou prsty nezaměnitelně užší, než předloktí, lze okolo dlaně ve vzdálenosti *offset_palm* od okraje vést čáru ve všech čtyřech směrech (viz obrázek 22). Následně se iteruje podél jedné hodnoty souřadnice (pro horizontální směr se jedná o souřadnici y) v každém směru a detekují změny v binárním obraze. Pokud jednotlivý nalezený objekt má šířku větší než je třetina šířky dlaně, jedná se o část předloktí a nezapočítává se. Zbylé objekty se zaznamenávají do počtu předpokládaných prstů. Pro přesnější detekci se daný algoritmus provede ještě jednou pro větší vzdálenost od dlaně. Pokud je pro obě vzdálenosti počet detekovaných prstů větší, než nula, je daný směr zaznamenán. Jelikož je potřeba znát i pozici palce, ukládají se dva směry prstů.

Empiricky nalezená nejlepší hodnota pro *offset_palm* je $\frac{5}{7}$ šířky dlaně, v druhém kole je zvolena dvojnásobná hodnota.



Obrázek 22 Zelený čtverec představuje hranice nalezené dlaně. Procházením podél žluté čáry, která je posunuta o offset se počítá počet změn a kontroluje tloušťka nalezených objektů. Pokud tloušťka odpovídá méně než třetině šířky dlaně, pak se považuje za prst.

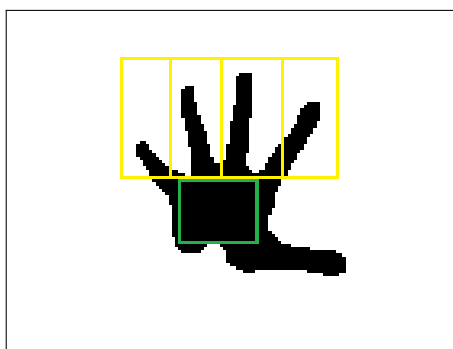
3.8.2 Detekce konců prstů

Všechny postupy se zakládají na prohledávání v okolí dlaně. Maximální délka prstů se odvozuje od nalezené velikosti dlaně. Prsty nesmí být ani příliš dlouhé ani příliš krátké.

Nejvýše položený bod

Nejjednodušší postup spočívá v rozdělení strany, na které se nachází prsty, na čtyři části, kde každá náleží jednotlivým prstům. Rozdělení je rovnoměrné a nepodporuje tak extrémní odchylky. Celkový obdélník musí být širší než dlaň a nebere se v potaz, který prst je v daném obdélníku nalezen. To znamená, že pokud se ukazují dva prsty, přičemž se některý/oba nachází ve vedlejším obdélníku, zaznamená se správně pouze počet prstů.

Jednotlivé části se následně procházejí a hledá se pixel, který ještě patří prstu a je nejvýše/nejniže dle orientace.



Obrázek 23 Zelený čtverec představuje nalezenou dlaň. Žluté obdélníky jsou předpokládané pozice prstů na základě nalezeného směru.

Nejvzdálenější bod

Další způsob rozšiřuje využitelnost předchozí metody. Stále se prochází oblast, ve které byly detekovány prsty, ale celkový obdélník se předá v kuse a hledá se největší vzdálenost

mezi nalezeným bodem a středem dlaně. Stále platí omezení délky, tudíž je přesnější největší možná vzdálenost, která stále patří prstu.

Nejvzdálenější bod dle pozice prstu

Funkce `findFingerTips3` si ukládá souřadnice, na kterých byly detekovány prsty z předchozí kapitoly (detekce směru prstů). Následně si dopočítá střed prstu a po přímce prohledává binární obraz tak daleko, dokud nenarazí na konec prstu (nulu v binárním obraze).

Nejvzdálenější bod z povoleného okolí

Další implementovaná možnost nalezení konců prstů spočívá v procházení oblasti, ve které se nachází prsty. Oblast je vyznačena žlutě na obrázku 23. Hledání začíná u vnější strany dané oblasti a jakmile se narazí na pixel patřící objektu, uloží se jeho pozice a souřadnice se zapíše do pomocného pole. Pole obsahuje souřadnice, na kterých se nachází již nalezené prsty a navíc jejich okolí, ve kterých se stále jedná o ten stejný prst. Při následujícím nalezení pixelu patřícího k objektu se tedy nejdříve zkontroluje, jestli se jedná o kolizi z pohledu již nalezených prstů.

3.8.3 Identifikace prstů

Index prstu se přidělí podle pozice špičky prstů vzhledem ke středu dlaně viz obrázek 23. Jelikož není známo, jakým směrem je ruka natočena ke kameře, není identifikovatelné, jestli se jedná o malíček nebo ukazováček. Indexy tedy představují pouze pozici prstu.

3.9 Vyhodnocení výsledků detekce gest

Vyhodnocení funkčnosti a použitelnosti navržených metod. TODO

4 Závěr

Vybraný postup implementace zaručil určité výhody oproti jiným algoritmům, které jsou zmíněné v kapitole 2.2. Mezi hlavní výhody patří, že není potřeba žádné jiné příslušenství, než je kamera. Člověk ovládající robota nemusí být oproti žádnému konkrétnímu pozadí, nemusí mít na ruku rukavice nebo mít jiné specifické náležitosti. Úvodní kalibrace také není nutná.

Jelikož je rozpoznávání vykonáváno v binárním obraze, je možná kombinace více programů nebo metod relativně jednoduchá. Zároveň lze program použít i s jinými hloubkovými kamerami. Pokud se algoritmus zkombinuje s využíváním obrazu z barevné kamery, lze detekci provádět spolehlivěji. Do programu lze také přidat nová gesta, odstranit stávající nebo změnit celkový pohled, na základě kterého se definují gesta. To lze udělat z pohledu přesnějšího rozpoznávání, tak i z pohledu, jak se jednotlivá gesta uvažují (počet prstů versus konkrétní prsty).

TODO zhodnocení výsledků - statistiky

Literatura

- [1] Jan Pikora. “Implementace grafických filtrů pro zpracování rastrového obrazu”. bakalářská práce. Masarykova univerzita, 2008.
- [2] Roger Boyle Milan Sonka Václav Hlaváč. *Image Processing, Analysis, and Machine Vision*.
- [3] *Skeletonization*. 17. květ. 2018. URL: <http://www.inf.u-szeged.hu/~palagyi/skel/skel.html#Top>.
- [4] Gunilla Borgefors. “Hierarchical Chamfer Matching: A Parametric edge matching algorithm”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1988).
- [5] Chikahito Nakajima a spol. “Full-body person recognition system”. In: *Pattern Recognition 36* (2003).
- [6] Yang Ran a Larry Davis Mohamed Hussein Wael Abd-Almageed. “Real-Time Human Detection in Uncontrolled Camera Motion Environments”.
- [7] Michael Van den Bergh. “Real-time 3D Hand Gesture Interaction with a Robot for Understanding Directions from Humans”. In: *IEEE International Symposium on Robot and Human Interactive Communication* (2011).
- [8] D Gavrilu. “Pedestrian detection from a moving vehicle”. In: *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II* (2000).
- [9] B. Winterstein a S. Müller G. Rigoll. “Robust Person Tracking in Real Scenarios with Non-Stationary Background Using a Statistical Computer Vision Approach”.
- [10] Christian von Hardenberg a François Bérard. “Bare-Hand Human-Computer Interaction”. In: *ACM Workshop on Perceptive User Interfaces*. 2001.
- [11] A. Erol et al. “Computer Vision and Image Understanding”. 2007.
- [12] Dieter W Fellner Hyosun Kim. “Interaction with Hand Gesture for a Back-Projection Wall”. 2003.
- [13] M. Abou Zliekha Noor Shaker. “Real-time Finger Tracking for Interaction”. In: *International Symposium on image and Signal Processing and Analysis*. 2007.
- [14] François Bérard Julien Letessier. “Visual Tracking of Bare Fingers for Interactive Surfaces”.
- [15] Rod McCall a Wolfgang Broll Georg Hackenberg. “Lightweight Palm and Finger Tracking for Real-Time 3D Gesture Control”.
- [16] *Microsoft Kinect used in rehabilitation of stroke patients*. 4. lis. 2011. URL: <http://metro.co.uk/2011/11/04/microsoft-kinect-used-in-rehabilitation-of-stroke-patients-208625/>.
- [17] Craig Feied a spol. “A Real-time gesture interface for hands-free control of electric medical records”. In: *AMIA Annu Symposium*. 2006.
- [18] Daniel Yong Wen Tan Ting Huang Yong. “Preliminary Investigation of Movement-Heart Rate Relationship Using Kinect-Based Badminton Performance Analysis”.

- [19] *bixi*. 1. květ. 2018. URL: <http://bixi.io>.
- [20] *Da-Jiang Innovations*. 7. květ. 2018. URL: <https://www.dji.com/>.
- [21] *heliguy*. 7. květ. 2018. URL: <https://www.heliguy.com/blog/2017/11/08/dji-intelligent-flight-modes>.
- [22] *openFrameworks*. URL: <http://openframeworks.cc> (cit. 13.10.2017).
- [23] *ofxKinectV2*. URL: <http://github.com/ofTheo/ofxKinectV2> (cit. 13.09.2017).
- [24] *Maximum size square sub-matrix with all 1s*. 28. břez. 2018. URL: <https://www.geeksforgeeks.org/maximum-size-sub-matrix-with-all-1s-in-a-binary-matrix/>.