

bakalářská práce

Hidden

Hidden



Červen 2018

Hidden

České vysoké učení technické v Praze
Fakulta elektrotechnická, Hidden

Poděkování

Text of acknowledgement...

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

Text abstraktu česky...

Klíčová slova

První klíčové slovo; druhé; třetí; ...

Abstrakt

Text of abstract in English...

Keywords

First keyword; second; third; ...

Obsah

1 Úvod	1
1.1 Důvody	1
2 Teoretická část	2
2.1 Stávající metody	2
2.1.1 Předzpracování	2
Binární obraz	2
Rozostření	2
2.1.2 Rozpoznávání postav	3
2.1.3 Rozpoznávání částí těla	5
2.1.4 Rozpoznávání gest	6
Rozpoznávání prstů	6
2.2 Aktuální aplikace	8
3 Praktická část	9
3.1 Software	9
3.1.1 openFrameworks	9
3.2 Zpracovatelné vstupy	9
3.2.1 Depth video	10
Úvodní nastavení	10
Nalezení ruky	10
Identifikace gesta	10
Vykreslení ruky	10
Nalezení ruky	10
Identifikace gesta	11
3.3 Omezení	11
3.3.1 Časová náročnost	11
4 Závěr	13
4.1 Možnosti využití gest při ovládání mobilního robotu	13
Literatura	14

Zkratky

Preliminary text...

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"... metoda podpůrných vektorů
ToF	"Time-of-Flight"... doba letu aneb hloubkové kamery založené na vý- počtu vzdálenosti z doby letu laseru
ROI	"Region of interest"... oblast zájmu
RBF	"Radial Basis Functions"
...	...

1 Úvod

V dnešní době robotizace, kdy lidé přicházejí na výhody využívání inteligentních (VLOŽIT REFERENCI podle jaké definice)

zařízení, je snaha rozšířit jejich využití co nejvíce.

POZN:vložit delší povídání o tom co se již děje atd?

Řízení zařízen za pomoci kamery se řadí 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.

1.1 Důvody

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 v reálném čase a jsou výpočetně složité a časové náročné. Cílem bakalářské práce je vytvořit program relativně jednoduše nastavitelná s tak rychlou odezvou, jak to jen bude robustnost dovolovat.

Následně řešené přístupy jsou voleny tak, aby měly co nejmenší odezvu. S lepší robustností již lze navázat na konci.

2 Teoretická část

2.1 Stávající metody

2.1.1 Předzpracování

Binární obraz

Většina metod se zakládá na zpracování binárního obrazu. Probíhá to zvolením prahové hodnoty, se kterou se všechny hodnoty porovnají a ve výsledku vznikne binární obraz. Objekt, který je středem zájmu, se skládá z pixelů s hodnotou 1 zatímco ostatní mají hodnotu 0.

Rozostření

Rozostření slouží jako relativně spolehlivá eliminace šumu v obraze. Nejosvědčenější metody jsou pomocí Gausse nebo mediánu.

Filtr medián se zaměřuje na vyhlazování lokálních extrémů z obrazu. Nejprve se seřadí všechny hodnoty z aktuální konvoluční matice a vybere hodnotu, která leží uprostřed seřazeného pole a tou nahradí zpracovávaný bod. Velikost okolí zahrnutého v analýze se může zvětšit při větší hustotě šumu [1]. Ale v aplikaci na hloubkový obraz, ve kterém se detekují prsty je důležité zachovat určité rozlišení.



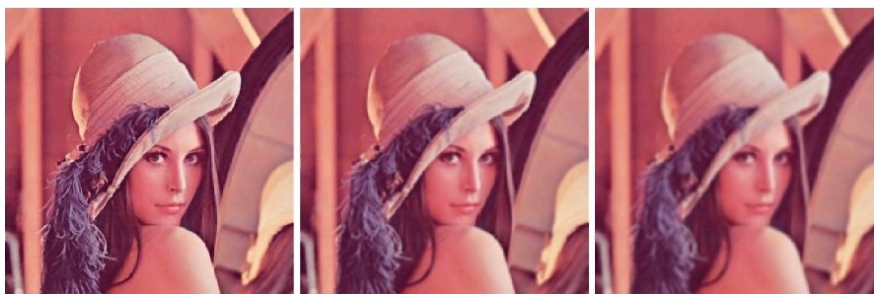
Obrázek 1 a) původní obraz b) průměrování c) medián Rozdíl mezi filtrem s průměrováním a mediánem [1]

Gaussův filtr je průměrování s Gaussovským rozložením [15]. Využívá se k vyhlazování obrazu a odstranění detailů a šumu.

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

Proti rozšíření do nekonečna se v konvoluční masce přidělí větší váha bodu ve středu. 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 Gaussův filtr [15]

2.1.2 Rozpoznávání postav

Možnosti:

podle tvaru

- na základě modelu
- na základě prvků
- strojově naučené programy
- neuronové sítě
- transformační matice vzdálenostních funkcí (chamfer)

podle barvy

- strojově naučené programy

Jednou z metod je dle Ch. Nakajima a spol. [2] ukládání několika obrázků, z nichž se detekuje pohybující se objekt a vše, co se nehýbe se označí za 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 použila SVM (support vector machines) a k-NN klasifikátor.

Hussein a spol. [3] používají k rozpoznání postavy prohledávání obrazu a porovnávání siluet s databází, ve které jsou vzory lidských postav. Shoda je detekována pomocí zkosením objektů. Vzory jsou ve formě binárního obrazu. Vzdálenost D mezi vzorem V a objektem z obrazu O se počítá vzorcem

$$D(O, V) = \frac{1}{|V|} \sum_i C_i V_i, \quad (3)$$

kde $|V|$ je počet pixelů v siluetě 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 co nejvíce na objekt a transformací bodů jednoho objektu pomocí parametrických transformačních rovnic.

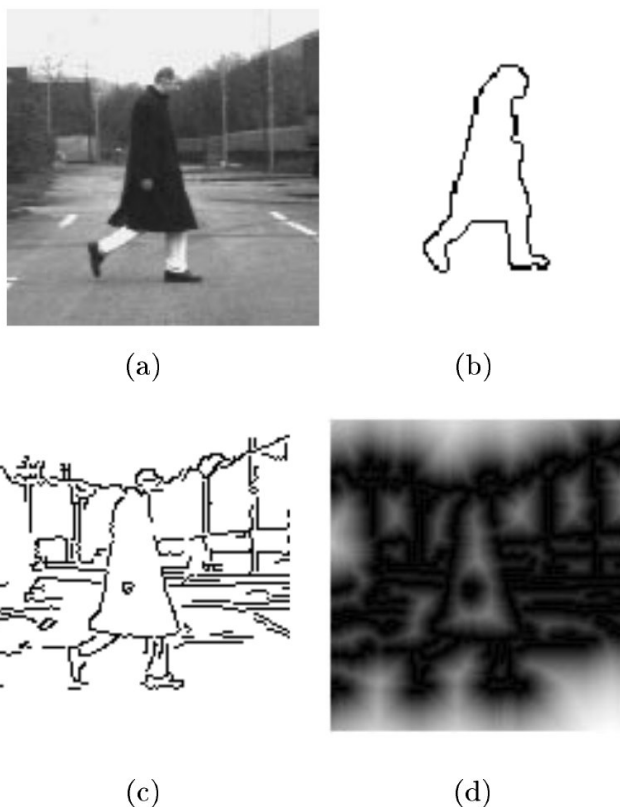
Jiný přístup poskytuje nalezení objektu v zorném poli pomocí hloubkové kamery, následným rozhodnutím, zda objekt má lidské nohy, nebo aspoň dva objekty odpovídající

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 definitivně nalezen člověk [4].

Gavrila [5] 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 v reálném čase. Shodu podle tvaru detekuje pomocí vzdálenosti křivek, která se počítá podobně jako u Hussein a spol. [3]

$$D(O, V) = \frac{1}{|V|} \sum_{t \in T} d_I(t), \quad (4)$$

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.

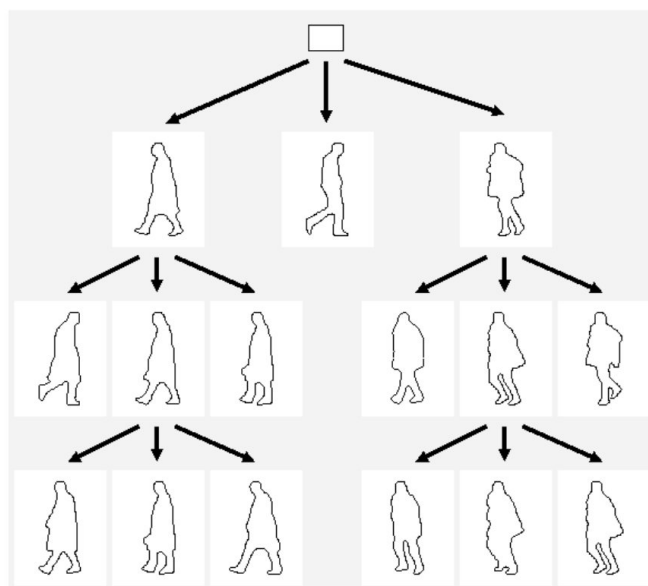


Obrázek 3 a) původní obraz b) vzor c) nalezené hrany d) vyobrazené vzdálenosti [5]

V rámci optimalizace se tato metoda (jak překládat chemfer matching system??) rozšiřuje na hierarchizovanou, která sdružuje podobné vzory do shluků a tudíž se prohledávání databáze zrychlí.

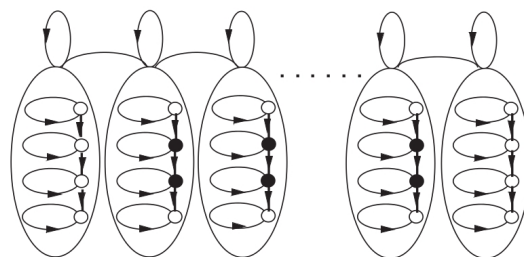
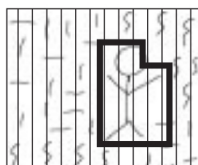
Ověření výsledku probíhá RBF (radial basis functions) klasifikátorem. Nejdříve se z původního obrazu vybere čtyřúhelník obsahující možnou postavu a na základě euklidovské vzdálenosti určuje, jestli jednotlivé pixely jsou součástí postavy či nikoliv.

Riggol a spol. [6] kombinují rozpoznávací metody založené na prvcích a na modelech za cílem dosažení spolehlivého a efektivního programu. Využívají P2DHMM (Pseudo-



Obrázek 4 využití hierarchie v databázi vzorů [7]

2D Hidden Markov Model). Jedná se o předem naučený algoritmus, který dokáže na základě pravděpodobností určit, jestli se jedná o postavu nebo ne.



Obrázek 5 stochastický model 2D objektu [11][6]

2.1.3 Rozpoznávání částí těla

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

Jestliže aplikace nevyžaduje dynamické využití, lze použít rozdíl mezi objektem a pozadím [14]. Vyžaduje to co nejvíce neměnné pozadí a dostatečný rozdíl mezi objektem a zbytkem. Rozdíl lze hledat jednoduše procházením pole nebo využitím "chytřejšího" algoritmu. Spočívá v identifikaci číste obrazu, která se hýbe oproti posledním obrazům.



Obrázek 6 a) vstup b) referenční obraz c) výstup ^[14]_[7]

2.1.4 Rozpoznávání gest

Mezi rozpoznávání na základě vzhledu se řadí mnoho metod zahrnujících strojově naučené algoritmy jako neuronové sítě, HMM (Skrytý Markovův model) a jiné [8]. Gesto lze identifikovat i pomocí extrakce (ČESKY) pozice a směru ruky a jednotlivých prstů.

Kim a spol. ^[5]_[9] 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 reality).

Shaker a Zliekha ^[12]_[10] získávají směr prstu na ruce, kterou snímají dvěma kamerami. Jedna kamera natáčí shora, zatímco druhá z boku. Oba vstupy se zpracovávají zvlášť do rozpoznávání gesta. 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 odstanění šumu se obrázek rozmaže a následně zaostří. V tomto kroce vzniká podmínka, že největší objekt v obraze je právě ruka. Poté se pomocí Laplaciána získají hrany daného objektu a z něj se vytvoří kontury. Gesto se identifikuje pomocí nalezení největšího vrcholu v obraze a nalezení čáry, kterou opisuje natažený prst a následnou kombinací obou směrů do 3D.

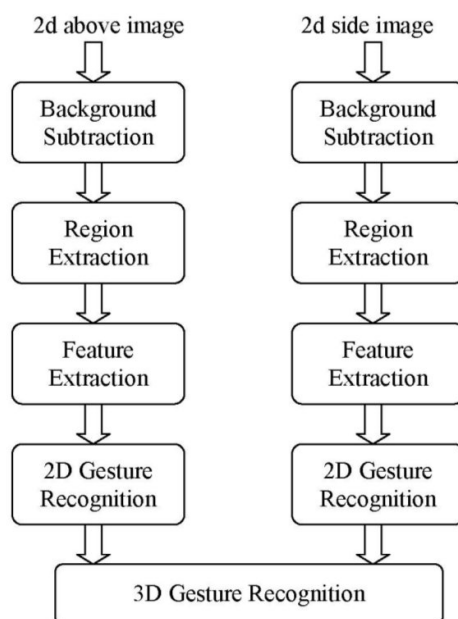
Rozpoznávání prstů

Hackenberg a spol. ^[12]_[10] implementovali identifikaci prstů a dlaně na základě několika kroků. Nejprve projde hloubkový obraz z kamery a hledá tvary podobné špičkám prstů nebo rourovitého tvaru podobného prstům. Poté se zaměří na detailnější vlastnosti dlaně. Z nalezených vhodných polí, která mohou představovat dlaně se vyřadí ta, která nejsou napojena nijak na prsty, 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 pomocí nalezení lokálního maxima v obrysu ruky. Ve všech adekvátních směrech se naleznou kandidáti a následně vyloučí mylné identifikace. Jedná se ale o postup náchylný k šumu v obraze, podobná metoda odolnější navrhuje porovnávání vzdálenosti obrysu k pozici ruky ^[3]_[8].

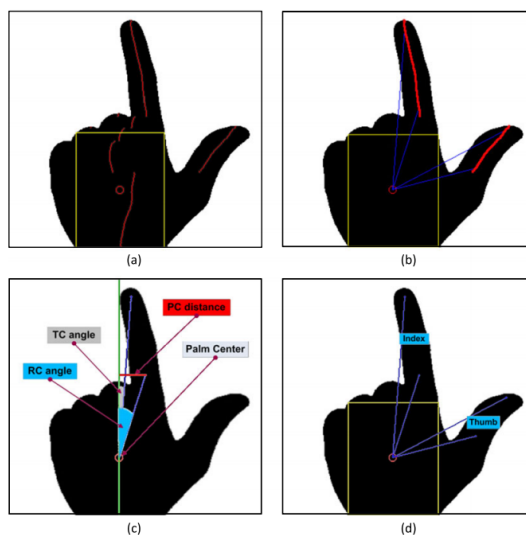
Jednotlivé prsty lze rozpoznat například pomocí vyhledávání na základě tvaru ^[4]_[11]. Jde o zjednodušení představy objektu na geometrický objekt a vyhledávání vhodných kandidátů v obraze.

Mezi další programy rozpoznávající špičky prstů z již relativně čistého čtyřúhelníku, který obsahuje zájmový objekt, patří identifikace na základě dvou specifických vlast-



Obrázek 7 popis algoritmu ^[12]_[10]

ností. Centrum špičky je obklopen kolem pixelů patřící prstu a ten je obklopen dlouhou řadou pixelů nepatřících prstu ^[14]_[7]. Ruce lze identifikovat i až po prstech a to tak, že se vyloučí objekty podobného tvaru (propisky, fixy), které nesplňují vlastnosti rukou, například nepatří k žádné dlani.



Obrázek 8 Vyobrazení definovaných pojmů, které lze využít k výpočtům pozic, na základě proporcionálních vlastností prstů. ^[13]_[12]

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

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

2.2 Aktuální aplikace

Všechny aplikace, které mohou využít výhod [14]:

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

Potenciál pro využití v:

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

Již využíváno v:

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

3 Praktická část

Some introductory text...

3.1 Software

3.1.1 openFrameworks

OpenFrameworks ^[13] je open source C++ nástroj pro kreativní programování. Využívá se doplněk ofxKinectV2 ^[14]. Oproti zabudovanému 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 `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 jen může. V `update` i `draw` se může využít funkce `ofGetTimef()`, která vrací vteřiny v jednotkách float od spuštění programu. Za využití modulu funkce (i jiných) můžeme ovlivnit jak často se bude vykonávat každá aktualizací část nebo její podčást.

3.2 Zpracovatelné vstupy

Kinect má dvě kamery, které poskytují pro účely projektu užitečné informace. Minimální vzdálenost pro správné měření je 0.5 m a maximální 4.5 m. Při větší vzdálenosti bude Kinect stále ještě detekovat věci v zorném poli, ale bude poskytovat nepřesná data a hloubková kamera pozbyde užitku. Velký vliv má i rozlišení, které je pro velké vzdálenosti nedostačující. V dále 5 m od kamery je rozlišení 7 cm.

Hloubková kamera poskytuje pole s hodnotami 0 až 255 dle vzorce pro výpočet vzdálenosti na základě doby letu infračerveného světla.

Stupnice hodnot je klesající, to jest nejbližší objekty mají hodnotu 255 (při experimentálních pokusech spíš 254). Ve vzdálenosti 3 metry hodnoty klesnou na 200.

UPŘESNIT A ZKONTROLOVAT JEŠTĚ NĚKOLIKRÁT

RGB kamera nabízí pole pixelů, kde každý má složky RGB (0 až 255). Dalo by se využít v kombinaci s robustním programem, který by na základě barvy, tvarů a dalších vylučovacích prvků správně detekoval nejdříve ruku a poté i gesta. Jedná se o znatelně náročnější způsob.

Odvoditelné parametry:

- Vzdálenost od kamery
- Vzdálenost mezi body
- Úhly

3.2.1 Depth video

Doporučený postup by měl sestávat ze čtyř částí:

Úvodní nastavení

Lze zde dát uživateli na vědomí podmínky interakce s kamerou. Například pozice ruky musí být v prostřední třetině. Alternativou může být počáteční umístění ruky do prostředí kamery, aby mohla dále vycházet z umístění za využívání historie nebo aby se zkrátila doba hledání ruky.

Nalezení ruky

Prostor pro určení podmínek a způsobu vyhledávání ruky. Tato část rozhoduje o robustnosti kódu a využitelnosti aplikace. Je vhodné nejdříve odstranit nulové a jinak nevyhovující pixely, vyplnit díry v obraze a počítat s dalšími chybami hardwaru i vlivy okolí. Pokud má být aplikace využívána v jakkoliv náročnějším okolí, musí obsahovat i vyloučení veškerých nevyhovujících objektů, jako jsou věci podobného tvaru či případně cizí ruce.

Identifikace gesta

Je vhodné prodiskutovat, jednotlivá gesta s jejich potenciálními uživateli, a rozhodnout se mezi dynamickými a statickými gesty. Případně by obě možnosti šly kombinovat, ale byl by to nejspíše zbytečně náročný postup. Pro gesta dynamická je potřeba udržovat si v paměti minulé stavy a stanovit dobu, pro kterou by to bylo přijatelné pro uživatele a zároveň to nemohlo být zaměnitelné nebo mylně identifikované. Nejjednodušší je identifikovat gesta statická, kde s využitím souřadnic lze vypočítat počet prstů, a tak v nejzákladnější implementaci bude prostor pro minimálně pět variant, pokud se vyloučí zavřená pěst pro zjednodušení. V momentě, kdy bude program podporovat identifikaci jednotlivých prstů od sebe, je k dispozici variant hned více, než by kdy bylo třeba nebo bybylo zapamatovatelné pro uživatele.

Vykreslení ruky

Objekty se přes video vykreslují pomocí FBO ("frame buffer object"). Jedná se o buffery s objekty, které je třeba vykreslit. Reprezentuje to plátno, na které se pomocí pomocí příkazů 'begin()' a 'end()' vykreslují 3D objekty a jednou za snímek či méně často (podle požadavků) se vykreslí pomocí příkazu 'draw()'. Pro lepší přehled se můžou jednotlivé prsty zvýraznit koulí, zatímco celá ruka krychlí. Vykreslení po obvodu se může zdát být zbytečným, jelikož by to vyžadovalo udržovat si v paměti celou množinu bodů po obvodu oproti jedinému bodu v případě jednotlivého objektu, který představuje celou ruku.

Nalezení ruky

Předpoklad pro následující text je zpracování videa ohledně nulových pixelů. Měly by být ignorovány v rámci chybného vstupu. Zároveň platí, že čím více podmínek se uživateli na začátku předloží, tím jednodušší je nalezení ruky.

U správně ukazovaného gesta se dá předpokládat stejná vzdálenost jednotlivých bodů od kamery. Mírné odchylky se dají buď zahrnout offsetem nebo vyloučit úplně, jelikož se dá předpokládat, že to nebude mít vliv na zpracování, pokud se jedná o krajní body.

Pokud by se vycházelo z předem definované vzdálenosti, šlo by nalézt největší možný čtverec, který by představoval dlaň. Pro účely nalezení by muselo existovat pole pixelů, které by reprezentovalo dlaň, či případně podobný objekt, který by se následně vyloučil podle dalších kritérií. Nadále by byla dlaň definovaná středem nalezeného čtverce a šířkou.

Následuje vylučování mylných objektů podle toho, jak robustní program má být.

U ruky se dá předpokládat, že obsahuje prsty, které mají proporcionálně ke dlani určitou velikost a vzdálenost od středu dlaně. Jelikož známe střed nalezené dlaně, lze identifikovat i prsty. Ze začátku musí být známo jaké gesto bude ruka zobrazovat při inicializaci, aby se dala nalézt. Pro ilustraci bude rozebráno gesto se všemi prsty nataženými.

Nejjednodušší metodou by bylo nalezení lokálních maxim po šíři dlaně. Když se dlaň rozdělí na čtyři části a v každé se nalezne maximum po ose y , pro každý prst se změří vzdálenost špičky prstu od středu dlaně. Pokud se bude výrazně odchylovat od proporcí běžné ruky, bude nalezený objekt vyloučen.

Promyšlenější programy využívají tvorbu skeletonu pro sledování a identifikaci jak postavy, tak i ruky. Jedná se o postup nalezení kloubů a jejich spojení, což pak reprezentuje ruku. Jedná se o možný, ale mírně náročnější a zdlouhavější postup.

Identifikace gesta

Z předpokladů uvedených výše lze vypočítat počet prstů, což je nejjednodušší způsob identifikace gesta. Pro přesnost se lze opřít i o úhly a vzdálenosti mezi středem dlaně a špičkami prstů podle vzorců.

REFERENCE na teoretickou část

Nejjistější by bylo procházet blok, který obsahuje ruku a nalézt počet prstů z ní vyčnívající. Jedná se ale o časově náročnější postup.

Sledování gesta může také probíhat pomocí ukládání historie. Za předpokladu, že se člověk hýbe pomaleji, než se střídají snímky k analýze, by se dala procházet menší část pole, například jen offset v okolí místa, kde se posledně nacházela ruka s prsty a kontrolovalo se, zda počet ukázaných prstů se zmenšil nebo zvětšil. K tomuto účelu by stále stačilo gesto definované počtem prstů bez závislosti na tom, o které se jedná.

Otázkou je i jaký objekt má představovat gesto. Jedná-li se o číslo, 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.

3.3 Omezení

3.3.1 Časová náročnost

Kamery, 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. Dále jsou uvedena data pro hloubkovou kameru.

Iterace samotným polem dvěma for cykly trvá zanedbatelných 0.8 ms, ale pokud při procházení načítáme hodnotu pixelu z kamery, časová náročnost je průměrně 5.8 s. Jelikož tato doba zpracování jediného snímku je nevhodná pro účely interakce s člověkem, musí se doba zkrátit.

Nejjednodušší, ale limitující způsob je procházet pouze prostřední třetinu, jelikož se tím zkrátí doba zpracování jednoho snímku na průměrných 0.65 s. Jelikož ani to není nikterak zázračný čas a často může být nedostatečný, je tento problém potřeba vyřešit.

Nabízí se několik možností jak ovlivnit periodu výpočtů, ale vzhledem k časové náročnosti čtení pixelů z videa se nejedá o řešení tohoto problému.

Program se snaží vykonávat všechny tři části tak často, jak může. Výpočetní analýzu můžeme provádět jednou za určitý čas. Aby to uživatel nezaznamenal, muselo by se jednat o desítky milisekund. V `update` i `draw` se může využít funkce `ofGetElapsedTimef()`, která vrací vteřiny v jednotkách `float` od spuštění programu nebo `ofGetElapsedTimeMillis()`, která vrací milisekundy od resetování čítače. Využitím modulu (i jiných) můžeme ovlivnit jak často se bude vykonávat každá aktualizací část.

Další způsob daného řešení je zpracovávat pouze některé snímky. Aplikace obsahuje funkci `isFrameNew()`, která vrací `boolean` a zjišťuje, jestli se snímek změnil či ne.

4 Závěr

4.1 Možnosti využití gest při ovládání mobilního robotu

Literatura

- [15] [1] Jan Pikora. “Implementace grafických filtrů pro zpracování rastrového obrazu”. bakalářská práce. Masarykova univerzita, 2008.
- [6] [2] Chikahito Nakajima a spol. “Full-body person recognition system”. In: *Pattern Recognition 36* (2003).
- [9] [3] Yang Ran a Larry Davis Mohamed Hussein Wael Abd-Almageed. “Real-Time Human Detection in Uncontrolled Camera Motion Environments”.
- [10] [4] 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).
- [7] [5] D Gavrilu. “Pedestrian detection from a moving vehicle”. In: *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II* (2000).
- [11] [6] B. Winterstein a S. Müller G. Rigoll. “Robust Person Tracking in Real Scenarios with Non-Stationary Background Using a Statistical Computer Vision Approach”.
- [14] [7] Christian von Hardenberg a François Bérard. “Bare-Hand Human-Computer Interaction”. In: *ACM Workshop on Perceptive User Interfaces*. 2001.
- [3] [8] A. Erol et al. “Computer Vision and Image Understanding”. 2007.
- [5] [9] Dieter W Fellner Hyosun Kim. “Interaction with Hand Gesture for a Back-Projection Wall”. 2003.
- [12] [10] M. Abou Zliekha Noor Shaker. “Real-time Finger Tracking for Interaction”. In: *International Symposium on image and Signal Processing and Analysis*. 2007.
- [4] [11] François Bérard Julien Letessier. “Visual Tracking of Bare Fingers for Interactive Surfaces”.
- [13] [12] Rod McCall a Wolfgang Broll Georg Hackenberg. “Lightweight Palm and Finger Tracking for Real-Time 3D Gesture Control”.
- [1] [13] *openFrameworks*. URL: <http://openframeworks.cc> (cit. 13.10.2017).
- [2] [14] *ofxKinectV2*. URL: <http://github.com/ofTheo/ofxKinectV2> (cit. 13.09.2017).