

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	Rozpoznávání postav	2
2.1.2	Rozpoznávání částí těla	2
2.1.3	Rozpoznávání gest	2
2.2	Aktuální aplikace	3
3	Praktická část	4
3.1	Software	4
3.1.1	openFrameworks	4
3.2	Zpracovatelné vstupy	4
3.2.1	Depth video	5
3.2.2	Úvodní nastavení	5
3.2.3	Nalezení ruky	5
3.2.4	Identifikace gesta	5
3.2.5	Vykreslení ruky	5
3.2.6	Nalezení ruky	5
3.2.7	Identifikace gesta	6
3.3	Časová náročnost	6
4	Závěr	8
4.1	Možnosti využití gest při ovládání mobilního robotu	8
	Literatura	9

Zkratky

Preliminary text...

OF	openFrameworks ... nástroj pro kreativní programování
FBO	"frame buffer object"... objekt pro vykreslování v OF
...	...

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 real-time zpracování a jsou výpočetně složité a časově 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

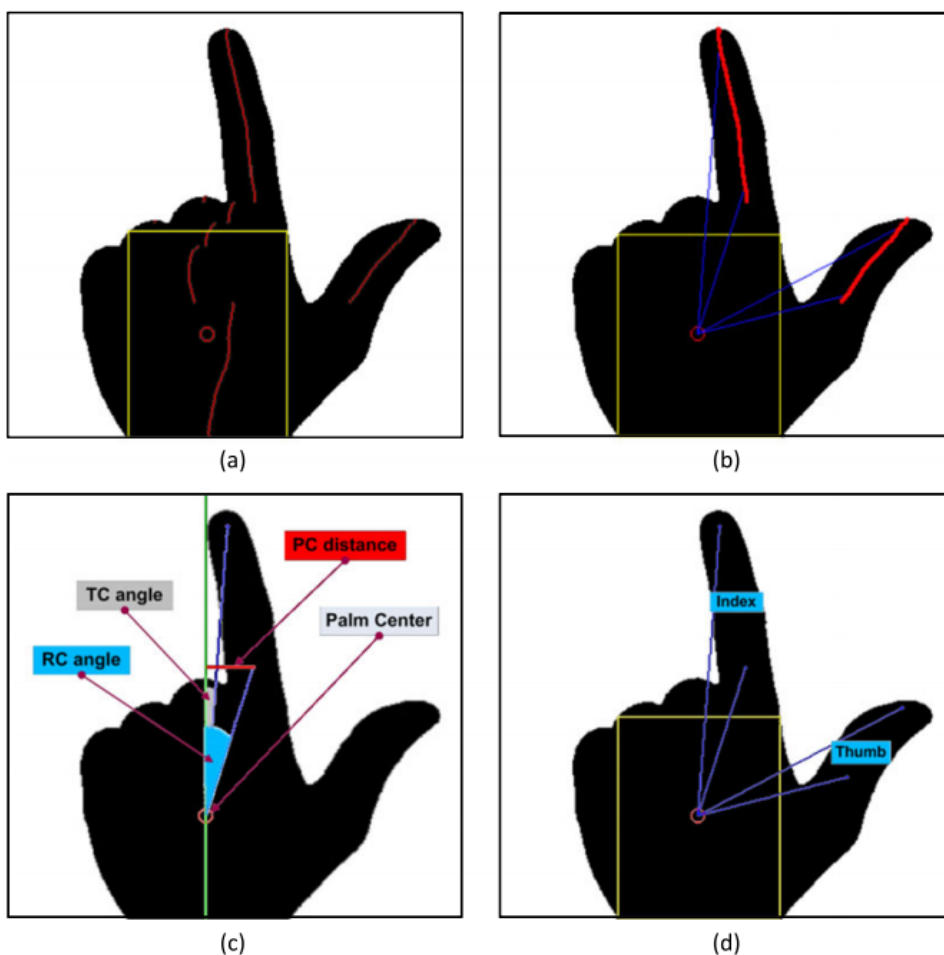
Some introductory text...

2.1 Stávající metody

2.1.1 Rozpoznávání postav

2.1.2 Rozpoznávání částí těla

2.1.3 Rozpoznávání gest



Obrázek 1 Vyobrazení definovaných pojmů, které lze využít k výpočtům pozic, na základě proporcionálních vlastností prstů. [1]

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

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

2.2 Aktuální aplikace

3 Praktická část

Some introductory text...

3.1 Software

3.1.1 openFrameworks

OpenfFrameworks ^[1]_[2] je open source C++ nástroj pro kreativní programování. Využívá se doplněk ofxKinectV2 ^[2]_[3]. 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 dálce 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 prostřed 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 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 zdlohavější 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 `modulo` (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

- 13

[1]

N. Papamarkos K. Sgouropoulos E. Stergiopoulou. *A Dynamic Gesture and Posture Recognition System*. 2014.
- 1

[2]

openFrameworks. URL: <http://openframeworks.cc> (cit. 13.10.2017).
- 2

[3]

ofxKinectV2. URL: <http://github.com/ofTheo/ofxKinectV2>.