

Realizace experimentů pro odhad parametrů dynamického modelu látky

Michal Neoral

2. dubna 2014

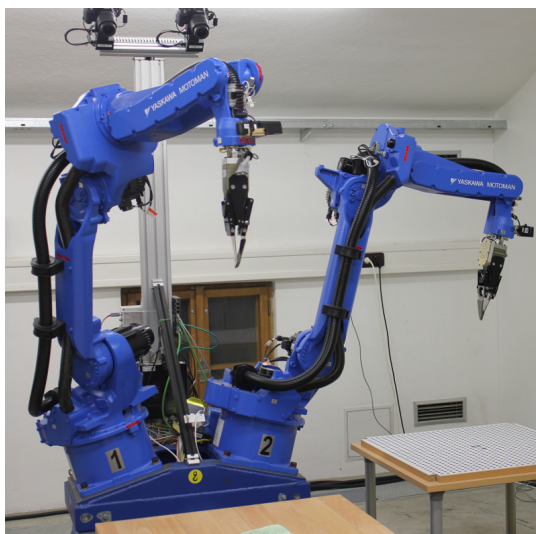
Kapitola 1

Úvod

1.1 O projektu

Tato práce je součástí mezinárodního projektu CloPeMa (Clothes Perception Manipulation). Tento dokument popisuje sběr dat a postup realizace experimentů pro odhad parametrů dynamického modelu textilie. Více informací o projektu CloPeMa na internetových stránkách projektu [1] a na wikipedii projektu [2]

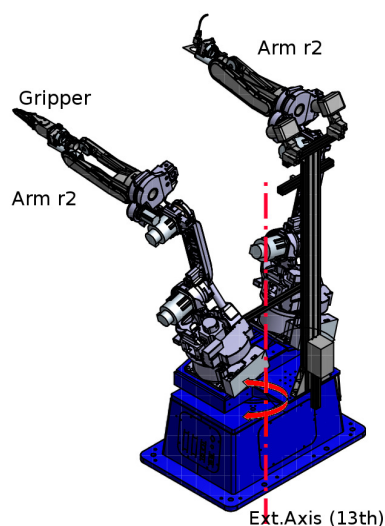
1.2 Popis pracoviště



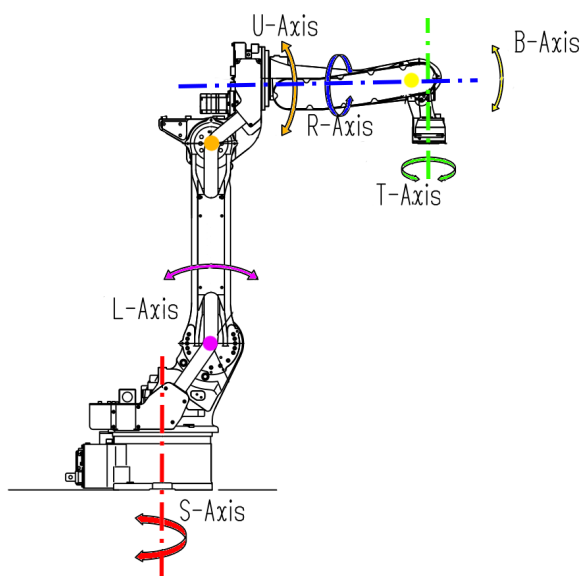
Obrázek 1.1: Manipulátor projektu CloPeMa umístěný na ČVUT v Praze.

1.2.1 Manipulátor

Základ manipulátoru tvoří dvě robotické paže Motoman MA1400. Paže jedna je označena jako $r1$ (nebo se také objevuje $R1$). Paže dvě je obdobně značena $r2$ ($R2$). Paže $r1$ a $r2$ jsou umístěny na otočném stole. Otočný stůl se otáčí kolem osy označované jako **Externí osa** (nebo také **Ext.** případně jako 13 osa). Umístění paží a otáčení **Ext.** osy lze lépe vyčíst z obrázku 1.2. Každá paže manipulátoru má 6 os, okolo kterých je schopna se otáčet. Osy jsou dle výrobce označeny písmeny S , L , U , R , T a B (obrázek 1.3). Toto označení nestačí a k písmennému názvu je třeba přiřadit i číslo paže, na které se tato osa nachází. Např.: osa S nacházející se na paži $r1$ se bude nazývat $S1$ apod. Obdobně jako u označení paží se můžeme setkat i s použitím malých písmen.



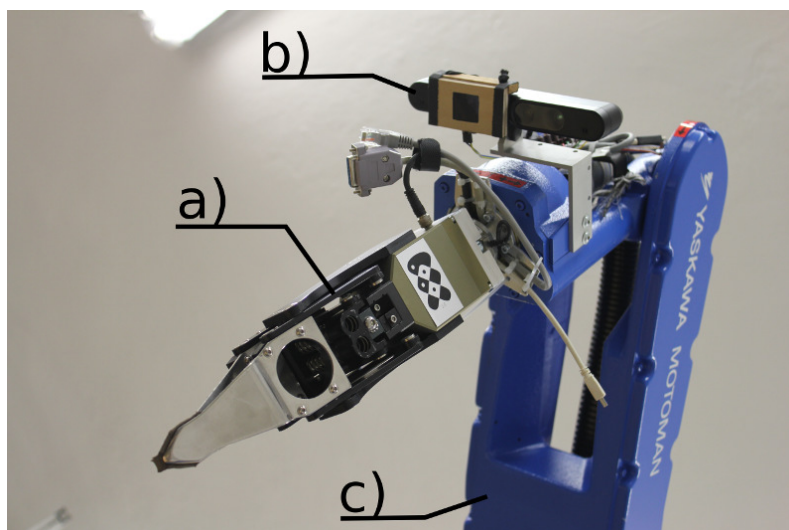
Obrázek 1.2: Označení ramen a umístění Externí osy.
 Arm r1 - paže r1, Arm r2 - paže r2, Gripper - chapadlo, Ext.Axis (13th) - externí (třináctá) osa.



Obrázek 1.3: Popis os robotické paže Motoman MA1400.
 S-Axis (červená) - osa S, L-Axis (fialová) - osa L, U-Axis (oranžová) - osa U, R-Axis (modrá) - osa R,
 B-Axis (žlutá) - osa B, T-Axis (zelená) - osa T.

1.2.2 Chapadlo

Každá z paží **r1** i **r2** je zakončena elektricky ovládaným chapadlem obrázek 1.4. Chapadla slouží k držení textilií.



Obrázek 1.4: Čapadlo.

a) čapadlo, b) kamera Asus Xtion, c) konec paže, na které je čapadlo namontováno.

1.2.3 Kamera

Další důležitou částí manipulátoru je 3D kamera Asus Xtion. Jedná se o kameru, která je schopná zaznamenávat jak RGB snímky¹, tak i hloubkovou mapu (`depth`). Kamera namontovaná na paži `r1` nese označení `xtion1`, kamera namontovaná na paži `r2` nese označení `xtion2`. Umístění kamery je patrné na obrázku 1.4.

Tento popis obsahuje pouze vybrané části, které jsou důležité pro tento experiment. Podrobnější popis robota naleznete na Wikipedii projektu CloPeMa [3].

1.3 Požadavky na experiment

Požadavkem na experiment je získání matematických příznaků, podle kterých by se daly odhadnout parametry dynamického fyzikálního modelu látky. Tyto příznaky chceme určit na základě sledování pohybu visící látky. Pohyb visící látky bude vyvolán pohybem čapadla manipulátoru, které látku drží.

Na základě senzorů, které máme k dispozici, jsme si zvolili:

- nejjednodušeji pohyb, o kterém si myslíme, že by nám mohl poskytnout potřebná data k získání parametrů dynamického modelu látky. Tímto pohybem je pohyb látky v rovině, ideálně vybuzeň pohybem čapadla s látkou po přímce (úsečce).
- dva typy sledování tohoto pohybu:
 - Standardní RGB kamerou sledujeme siluetu, případně samotnou látku proti stálému pozadí při pohybu **kolmo k optické ose**.
 - Senzorem snímající hloubkovou mapu sledujeme látku při pohybu **podél optické osy**.

¹Pro lepší manipulaci s RGB snímky je vhodné odfiltrovat pozadí. Způsobu provedení tohoto odfiltrování je naznačen v kapitolách 2.2.2 a 5.

Kapitola 2

Způsob pořízení dat

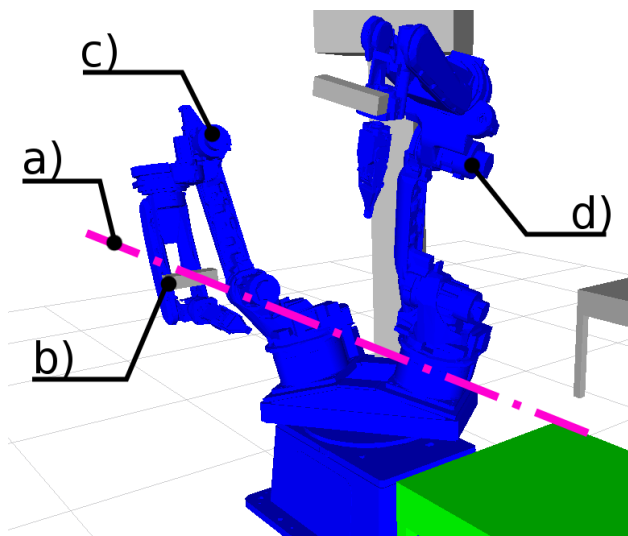
2.1 Realizace

Už při prvních pokusech jsme zjistili, že dynamika manipulátoru není natolik rychlá, aby s ním bylo možné provádět požadovaný pohyb chapadlem s látkou po úsečce potřebnou rychlostí (kapitola 1.3). Je ovšem možné dosáhnout potřebné rychlosti, jestliže pohyb bude vycházet pouze z jednoho kloubu. Proto jsme s omezením robota museli pohyb chapadla s látkou realizovat tak, že pohyb chapadla po úsečce je aproximován pohybem chapadla po části kružnice. Dalším omezením je prostorové omezení a to takové, že není možné umístit kameru `xtion` do pozice vhodné ke snímání RGB kamerou (tedy do pozice, kdy se chapadlo s látkou pohybuje kolmo k optické ose) a následně kameru `xtion` přesunout do pozice vhodné ke snímání hloubkové mapy (tedy do pozice, kdy se chapadlo s látkou pohybuje podél optické osy). Toto omezení jsme vyřešili pomocí toho, že pozice kamery `xtion`, tedy pozice paže s kamerou, je neměnná pro snímání záznamu RGB i pro snímání hloubkové mapy. Místo toho provede paže s textilií požadovaný pohyb chapadlem dvěma různými způsoby tak, aby byly splněny správné podmínky pro snímání jednotlivými senzory (kap. 1.3 - pozice kolmo vs. podél optické osy).

2.2 Výchozí pozice paží a celého manipulátoru

2.2.1 Paže s kamerou

Záznam je pořízen kamerou `xtion1` namontovanou na paži `r1`. Paže `r1` najede do polohy, ve které směřuje optická osa kamery `xtion1` vodorovně. Zároveň je optická osa kamery `xtion1` orientovaná směrem k paži `r2` (obrázek 2.1).



Obrázek 2.1: Pozice paže s kamerou.

a) Optická osa kamery `xtion1`, b) Kamera `xtion1`, c) paže `r1`, d) paže `r2`.

2.2.2 Paže s textilií

Textilie je držena pomocí chapadla namontovaného na paži **r2**. Paže **r2** nabývá dvou základních poloh.

Poloha pro měření

Paže **r2** je v poloze, ve které je připravena k provedení experimentu. Paže **r2** drží textilií v chapadle. Paže **r2** je ve výšce, ve které je kamerou **xtion1** zaznamenána textilie. Paže **r2** je také v pozici, aby mohla provádět pohyb potřebný pro experiment (kapitola 1.3 a kapitola 2.3).

Poloha pro referenční snímek

Tato poloha slouží k záznamu referenčního snímku pozadí, který slouží k odfiltrování pozadí z RGB snímků pro zlepšení přesnosti výsledků experimentu. Referenční snímek pozadí se nasnímá tak, že paže **r2**, v jejímž chapadle je držena látka, změní pozici tak, aby byla zcela mimo oblast záznamu kamery **xtion1**. V této pozici se provede záznam pozadí a paže **r2** s látkou se následně vrátí zpět do polohy pro měření. Více se odfiltrování pozadí budu věnovat v kapitole 5.4.1.

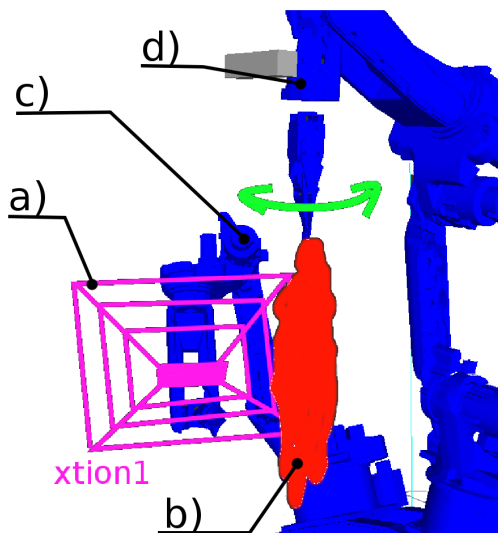
2.2.3 Otočení Ext.osy

Externí osa (osa č.: 13) je otočena tak, aby v pozadí snímané textilie bylo co nejméně rušivých předmětů. Nejlépe jednobarevný rovný povrch.

2.3 Pohyby paží

2.3.1 Pohyb paže, aby se látka pohybovala kolmo k optické ose

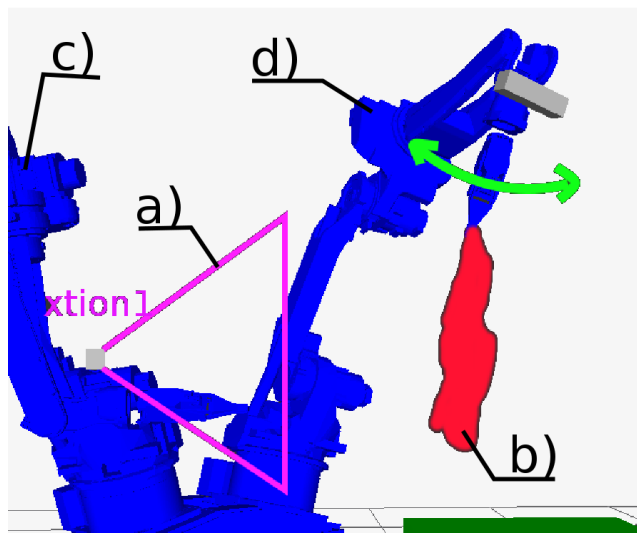
Paže **r1** neprovádí žádný pohyb a je v poloze popsané v kapitole 2.2.1. V chapadle paže **r2** je držena textilie. Paže **r2** provede požadovaný pohyb s touto textilií tak, že se otočí okolo osy **B** o určitý úhel a vrátí se zpátky do výchozí polohy. Pro lepší popsání pohybu je pohyb naznačen i na obrázku 2.2. Tento pohyb je vhodný ke snímání RGB kamerou.



Obrázek 2.2: Nastínění pohybů chapadla s textilií kolmo k optické ose.
a) naznačení zorného pole kamery **xtion1**, b) textilie, c) paže **r1**, d) paže **r2**.

2.3.2 Pohyb paže tak, aby se látka pohybovala podél optické osy

Paže **r1** neprovádí žádný pohyb a je v poloze popsané v kapitole 2.2.1. V chapadle paže **r2** je držena textilie. Paže **r2** provede požadovaný pohyb s touto textilií tak, že se otočí okolo osy **R** o určitý úhel a vrátí se zpátky do výchozí polohy. Pro lepší popsání pohybu je pohyb naznačen i na obrázku 2.3. Tento pohyb je vhodný ke snímání hloubkové mapy.



Obrázek 2.3: Nastínění pohybů chapadla s textilií podél optické osy
a) naznačení zorného pole kamery xtion1, b) textilie, c) paže r1, d) paže r2.

Přesné polohy a pohyby paží je možné vyčíst ze souboru:

`path_to_workspace/clopema_collect_model_data/src/_pos.py`. Pro nahlédnutí do tohoto souboru je nutné splnit předpoklady z kapitoly 3.1.

Kapitola 3

Pořízení dat

3.1 Předpoklady

1. Nainstalovaný ROS Hydro a balíčky CloPeMa dle návodu z Wikipedie projektu [4].
2. V balíčku `clopema_cvut/clopema_collect_model_data` ve složce `src` uprav script `local_options.py` pro místní nastavení počítače. Po otevření v příslušném textovém editoru upravte obsah proměnných pro Vaše umístění.
 - `pcglocate` pro umístění složky se zdrojovými kódy
 - `savefolder` umístění složky, kam se budou ukládat získaná data.

3.2 Postup spouštění

1. Před měřením nastavte omezení rychlosti robota na TEACH-PENDANTu na 10000 dle návodu [5]. Po měření uveďte nastavení omezení do původního stavu.
2. Spustíme robota:
 - `roslaunch clopema_launch start_robot.launch`
3. Spustíme kameru na paži r1:
 - `roslaunch clopema_launch xtion1.launch`
4. Poté, co se body 2 a 3 úspěšně spustí můžeme spustit vlastní skript pro sběr dat:
 - `roslaunch clopema_collect_model_data collect_data.py`
5. Program je ovládán z příkazové řádky.

3.3 Ovládání skriptu `collect_data.py` pro snímání dat

3.3.1 Popis ovládání

Skript již máme spuštěn podle návodu v kapitole 3.2. Po spuštění se nám zobrazí úvodní menu zobrazené na obrázku 3.1. Čísla v závorkách (**číslo**) značí pozici funkce v menu programu. Např. funkci (**4**) *Open Gripper* (otevření chapadla) spustíme vepsáním čísla 4 a potvrzením klávesy **Enter**. Obdobně postupujeme při spouštění i dalších funkcí. Pro ukončení programu vepíšeme slovo **stop** a potvrdíme klávesou **Enter**. Program **nedoporučuji** ukončit stiskem `ctrl+c`.


```
...1...Move to the HOME position (home)
...2...Move to READY TO MEASURE position (mpos)
...3...Move and record (action)
...4...Open Gripper (open)
...5...Close Gripper (close)
...6...Camera default record (camdef)
..stop..EXIT
```

Obrázek 3.1: Náhled menu skriptu

3.3.2 Postup pro nasnímání obrazu pro odfiltrování pozadí

1. Využijeme funkci **(6)** – *Camera default record* (obrázek 3.1).

3.3.3 Postup pořízení dat – manuální vkládání textilie:

1. Umístíme robota do polohy, ve které je připraven k měření **(2)** - *Move to READY TO MEASURE position* (obrázek 3.1).
2. Otevřeme chapadlo **(4)** – *Open Gripper*.
3. Zavřeme chapadlo **(5)** – *Close Gripper*. Po stisknutí máme 5 vteřin pro vložení textilie do chapadla než se chapadlo sevře.
4. Po sevření chapadla ustoupíme do bezpečné vzdálenosti od robota.
5. Zahájíme měření a záznam **(3)** - *Move and record*.
6. Budeme vyzváni k pojmenování souboru. Doporučuji nazývat souboru názvem textilie, případně i pořadovým číslem.
7. Po schválení názvu souboru bude provedeno měření způsobem, který je popsán v předchozí kapitole Způsob pořízení dat (kapitola 2).
8. Postup 2. až 7. můžeme opakovat pro další měření.
9. Před ukončením programu můžeme pomocí **(1)** umístit robota do výchozí polohy.
10. Program ukončíme pomocí **(exit)**.

Kapitola 4

Uložení dat

4.1 Formát dat

Data se ukládají pomocí rosbag (nástroj ROSu) ve formátu “.bag“ do předem určené složky uložené v souboru `local_options.py` (`path_to_workspace/clopema_cvut/clopema_collect_model_data/src/local_options.py`).

4.2 Témata (topics)

Z důvodu úspory místa a kapacity přenosového kanálu jsou zaznamenány pouze témata (topics), která jsou uložena v souboru `path_to_workspace/clopema_cvut/clopema_collect_model_data/matlab/topics/topics.txt`. Pro tento experiment jsem vybral tyto témata (topics):

```
/joint_states
/tf
/xtion1/depth/camera_info
/xtion1/depth_registered/camera_info
/xtion1/projector/camera_info
/xtion1/rgb/camera_info
/xtion1/depth/image_raw
/xtion1/rgb/image_raw
/xtion1/depth/disparity
```

Seznam témat je možné libovolně měnit. Zaznamenáno je 7 vteřin dat.

4.3 Formát názvu

Zaznamenané soubory jsou ve tvaru: `name_speed_AX.bag`

- name vámi zadaný název
- speed nastavená rychlost manipulátoru
- A osa, kterou byl vykonán pohyb
 R nebo B (obrázek 1.3).
- X číslo souboru s tématy, která jsou zaznamenána

Kapitola 5

Načtení dat pro další zpracování

5.1 Předpoklady

1. Stáhnuty balíčky CloPeMa dle návodu z Wikipedie projektu [4].
2. Nainstalujte Matlab (odzkoušeno ve verzi 2012b i 2013a).
3. Nainstalujte toolbox [6] pro matlab „rosbag“ a přidána cesta pro tento toolbox.
4. Nacházejte se ve složce se zdrojovými kódy pro Matlab.
5. Soubor `topics/topics.txt` musí být stejný jako v době nahrávání .bag souboru.
6. Soubor `local_options.m` přizpůsobte vašemu nastavení (blíže popsáno přímo v souboru po otevření souboru v editoru).

5.2 Načtení souborů do Matlabu

1. Spusťte skript `startup.m`. Skript připraví prostředí a načte proměnné ze souboru `local_options.m`.

- `startup`

2. Pomocí funkce `loadBagFile()` načtěte požadovaný .bag soubor.

- `msgs = loadBagFile(path_to_bag_files, topics, bagfile)`

Vstupy do funkce

- `path_to_bag_files` (*řetězec*) - obsahuje umístění složky ve které jsou uloženy .bag soubory.
- `topics` (*matice buněk s řetězci*) - obsahuje názvy témat (topic), která se mají vybrat z .bag souboru pro zpracování.
- `bagfile` (*řetězec*) - název požadovaného .bag souboru. Způsob nazývání souborů byl popsán v kapitole 4.3.

Výstupy z funkce

- `msgs` (*matice buněk*) - obsahuje témata (topic) z požadovaného .bag souboru (data a informace).

3. Pomocí funkce `loadBackgroundRGB()` načtěte referenční snímek pro odfiltrování pozadí při snímání RGB kamerou.

- `rgb_back = loadBackgroundRGB(path_to_bag_files, bagfile_background)`

Vstupy do funkce

- `path_to_bag_files` (*řetězec*) - obsahuje umístění složky ve které jsou uloženy .bag soubory.
- `bagfile_background` (*řetězec*) - název požadovaného .bag souboru s nasnímaným pozadím. Název pro výchozí souboru je `camera_default_0.bag`.

Výstupy z funkce

- `rgb_back (matice)` - obsahuje RGB obrázek pozadí ve formátu *double*. Možno zobrazit pomocí `image(rgb_back)`.

Funkce z bodu 2 a 3 lze také spustit pomocí příkazu `loader`.

5.3 Předzpracování dat

Jak jsem uvedl v kapitole 2.3 a 4.3 je záznam pohybu chapadla s látkou rozdělen do dvou souborů. Záznamy končící písmenem **R** jsou určeny pro následné zpracování dat ze senzoru hloubkové mapy a záznamy končící písmenem **B** jsou určeny pro zpracování RGB snímků (písmena označují, kterou osou je na daném záznamu pohybováno pro vytvoření pohybu látky). Další zpracování se tedy bude lišit podle toho, zda chceme extrahovat matematické příznaky látky z hloubkové mapy (5.5) nebo z RGB snímků (5.4).

5.4 Zpracování RGB dat

Z kapitoly 5.2 již máme připravená data v matici buněk `msgs`. Z této matice buněk extrahujeme pouze data, která se týkají RGB obrázku a vytvoříme si posloupnost nasnímaných RGB obrázků s časy ve kterých byly zachyceny. Posloupnost vytvoříme pomocí funkce `makeFrontOfRGB()`. Na obrázku 5.1 můžete vidět příklad použití této funkce.

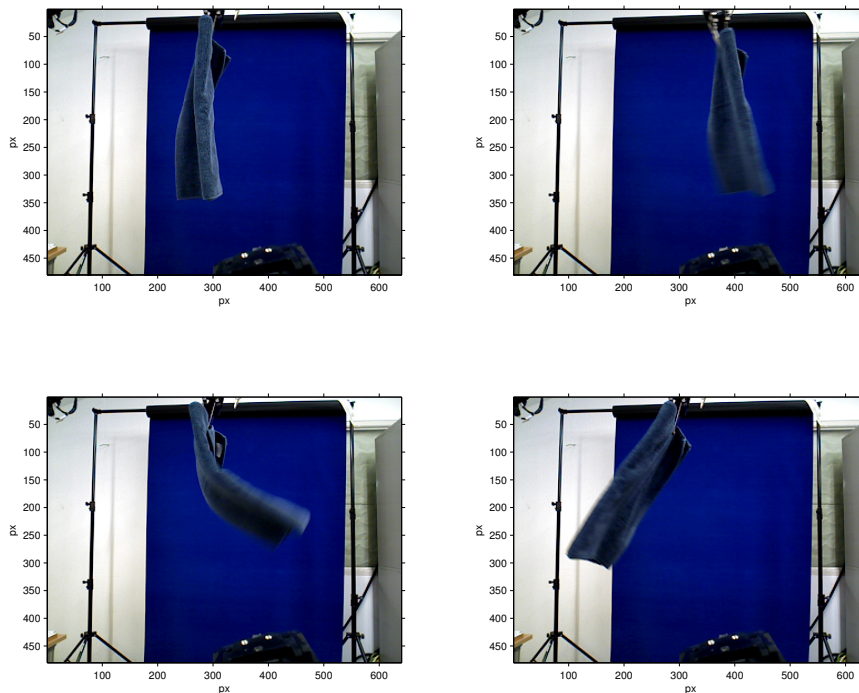
- `frontOfRGB = makeFrontOfRGB(msgs)`

Vstupy do funkce

- `msgs` (*matice buněk*) - je výstup z funkce popsané v 5.2.2.

Výstupy z funkce

- `frontOfRGB` (*matice buněk*) - obsahuje posloupnost RGB obrázků (*double* 640x480x3) a čas zachycení obrázku v *s* (*double*).



Obrázek 5.1: Příklad zaznamenaných RGB snímků.

5.4.1 Filtrování RGB snímků pomocí referenčního snímku

Na RGB snímcích jsou mi pohybující se látka i věci nedůležité pro experiment (stále pozadí, části manipulátoru...). Tyto části obrazu je potřeba odstranit. Odstraněním části obrazu mimo pohybující se látku nám pomůže snáze sledovat pohybující se siluetu látky. Látku odstraníme pomocí referenčního snímku, jehož způsob pořízení jsme si uvedli v kapitolách 2.2.2 a 3.3.2. Funkce `filtrReferenceFrame()` slouží k tomuto filtrování. Na obrázku 5.2 a 5.3 můžete vidět příklad použití této funkce.

- `[filtredFrontOfRGB, maskFrontOfRGB] = filtrReferenceFrame(frontOfRGB,referenceFrame,sizeMorpMask)`

Vstupy do funkce

- `frontOfRGB` (*matice buněk*) - matice buněk obsahující obrázky.
- `referenceFrame` (*matice double 640x480x3*) - obsahuje referenční snímek.
- `sizeMorpMask` (*integer*) - *nepovinný* parametr, kterým se nastavuje velikost masky pro morfologickou operaci "opening" [10].

Výstupy z funkce

- `filtredFrontOfRGB` (*matice buněk*) - obsahuje filtrované obrázky RGB a časovou značku původních snímků.
- `maskFrontOfRGB` (*matice buněk*) - obsahuje filtry obrazů RGB a časovou značku původních snímků.

Případně lze použít k dalším výpočtům pouze masku a není nutné načítat i filtrovaný obrázek. Funkce `maskReferenceFrame()` slouží k tomuto filtrování. Na obrázku 5.3 můžete vidět příklad použití této funkce.

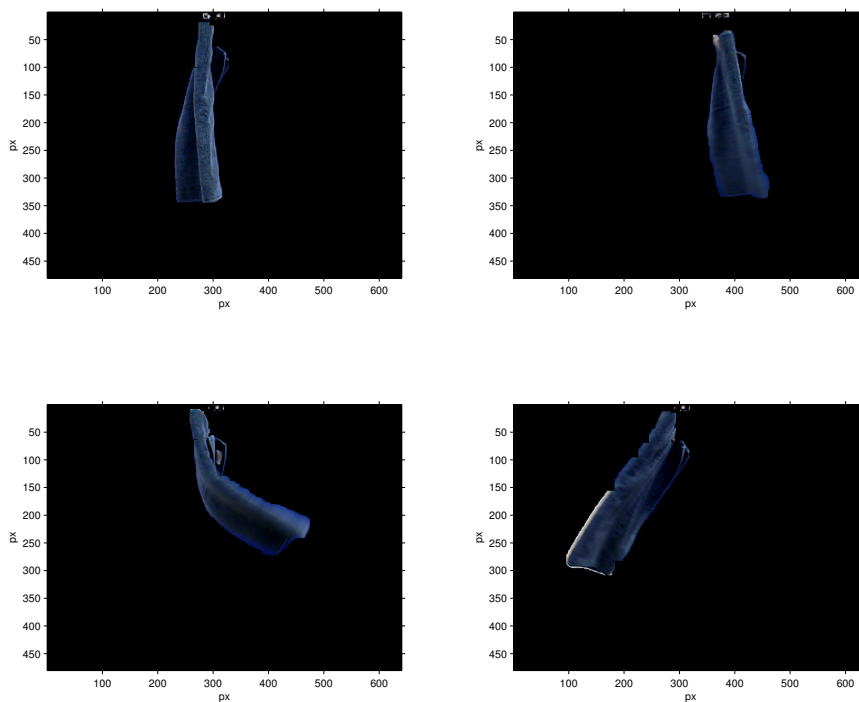
- `[maskFrontOfRGB] = maskReferenceFrame(frontOfRGB,referenceFrame,sizeMorpMask)`

Vstupy do funkce

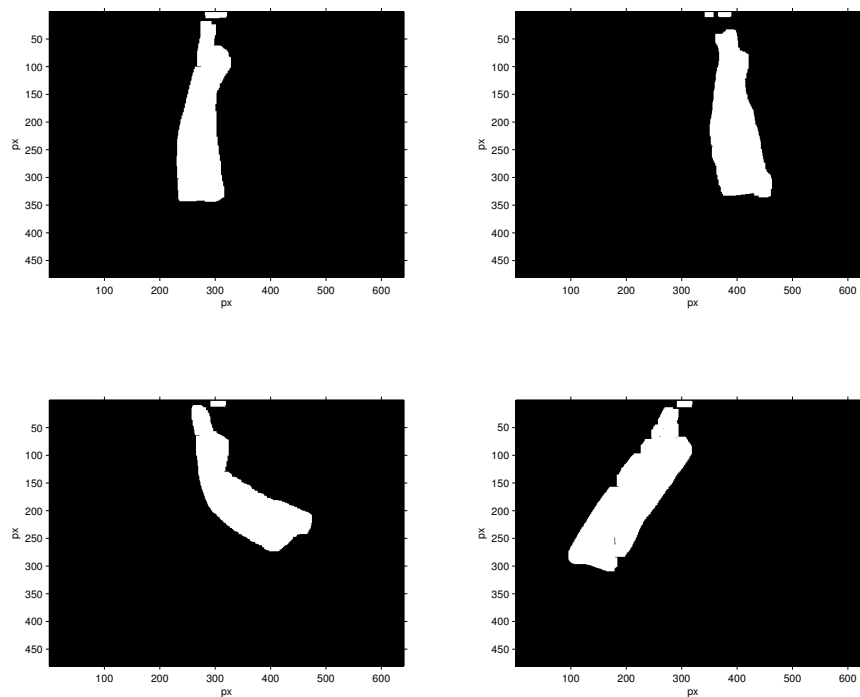
- `frontOfRGB` (*matice buněk*) - matice buněk obsahující obrázky.
- `referenceFrame` (*matice double 640x480x3*) - obsahuje referenční snímek.
- `sizeMorpMask` (*integer*) - *nepovinný* parametr, kterým se nastavuje velikost masky pro morfologickou operaci "opening" [10].

Výstupy z funkce

- `maskFrontOfRGB` (*matice buněk*) - obsahuje filtry obrazů RGB a časovou značku původních snímků.



Obrázek 5.2: Příklad vyfiltrovaných RGB snímků.



Obrázek 5.3: Příklad vyfiltrovaných RGB snímků - maska.

5.4.2 Nalezení kostry obrazů

Na obrázku 5.4 můžete vidět příklad použití této funkce.

- `[centerOfFabric] = findCenterOfFabric(maskFrontOfRGB)`

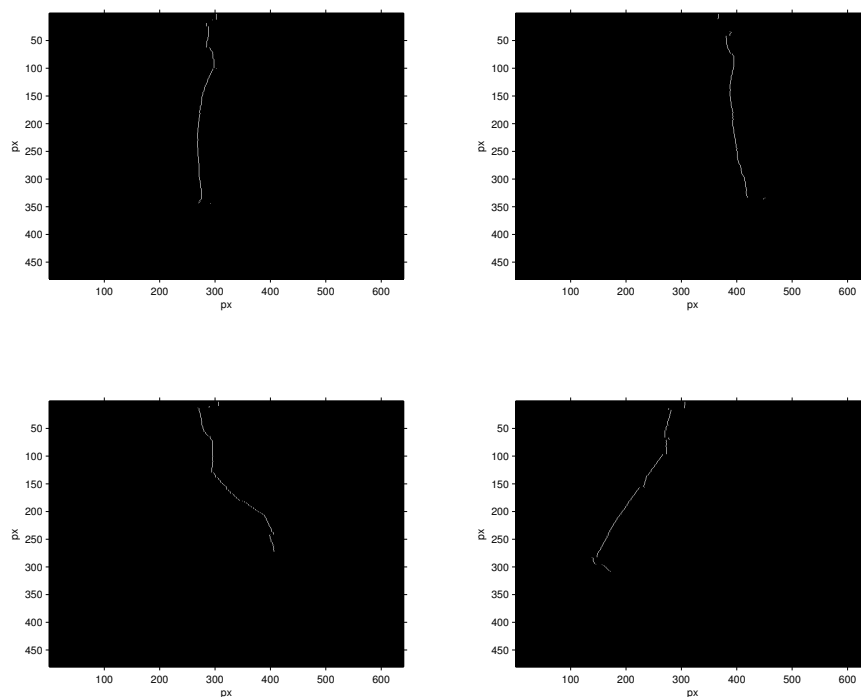
Vstupy do funkce

- `maskFrontOfRGB` (*matice buněk*) - obsahuje filtry obrazů RGB a časovou značku původních snímků.

Výstupy z funkce

- `centerOfFabric` (*matice buněk*) - obsahuje x-ové souřadnice středů látky získaných z filtrové masky v jednotlivých snímcích a časy zachycení snímku.

Zde vložit popis



Obrázek 5.4: Příklad středu látky v obraze.

5.4.3 Získání matematických příznaků látky z kostry obrazů v čase

Na obrázku 5.5 a 5.6 můžete vidět příklad použití této funkce.

- `[struct_points] = makePointsFromCenter(centerOfFabric, n_points, timeOfStart)`

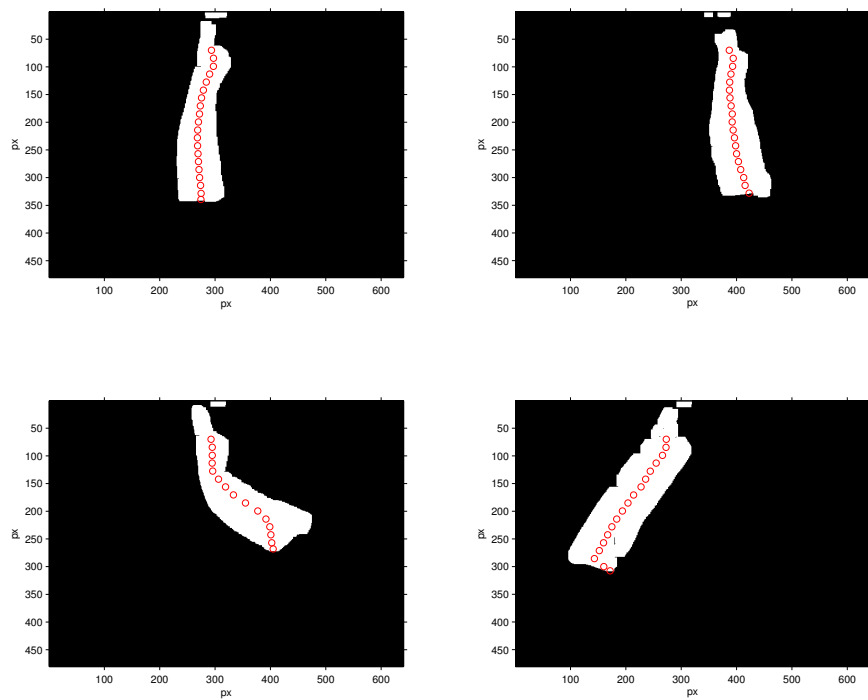
Vstupy do funkce

- `centerOfFabric` (*matice buněk*) - obsahuje x-ové souřadnice středů látky získaných z filtrové masky v jednotlivých snímcích a časy zachycení snímku.
- `n_points` (*integer*) - obsahuje požadovaný počet bodů, na který má funkce pohybující se látku rozdělit.
- `timeOfStart` (*double*) - *nepovinný* parametr, který značí začátek pohybu v čase [s] (*default=0*).

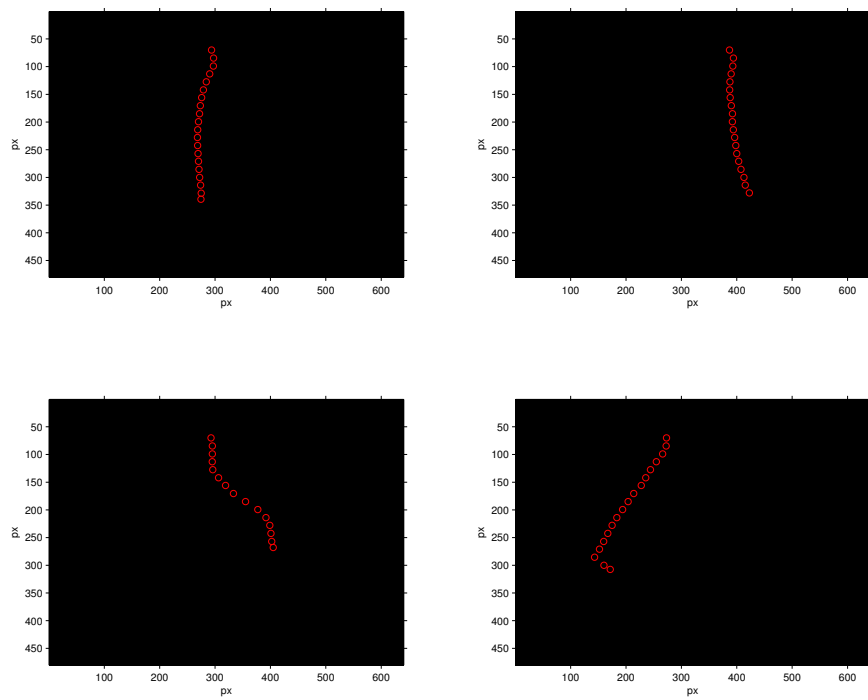
Výstupy z funkce

- `struct_points` (*matice buněk*) - obsahuje n bodů [y,x] (*dle n_points v každém snímku a čas záznamu snímku*).

Zde vložit popis



Obrázek 5.5: Příklad masky s nalezenými body.



Obrázek 5.6: Příklad nalezených bodů.

5.5 Zpracování dat ze senzoru hloubkové mapy

Z kapitoly 5.2 již máme připravená data v matici buněk `msgs`. Z této matice buněk extrahujeme pouze data, která se týkají hloubkové mapy a vytvoříme si posloupnost nasnímaných hloubkových map s časy ve kterých byly zachyceny. Posloupnost vytvoříme pomocí funkce `makeFrontOfDepth()`.

- `frontOfDepth = makeFrontOfDepth(msgs)`

Vstupy do funkce

- `msgs` (*matice buněk*) - je výstup z funkce popsané v 5.2.2.

Výstupy z funkce

- `frontOfDepth` (*matice buněk*) - obsahuje posloupnost hloubkových map (*double* 640x480) a čas zachycení mapy v *s* (*double*).

Zde vložit obrázky - příklad zobrazení hloubkové mapy zachycené senzorem

5.5.1 Filtrování hloubkové mapy dle vzdálenosti zájmové oblasti

Zde vložit popis

Zde vložit obrázky - odfiltrování nežádoucích vzdáleností

5.5.2 Nalezení bodů (oblastí) na povrchu látky

Nalezení oblastí výskytu pohybující se látky

- `nanmask = depthNanFilter(frontOfDepth)` Vstupy do funkce
 - `XXX (XXX)` - `XXX`.

Výstupy z funkce

- `XXX (XXX)` - `XXX`.

Zde vložit popis

Zde vložit obrázky - příklad nalezených bodů

Vzor bodů

- `pointPattern = depthPointsPosition(nanmask, n_points_x, n_points_z, otherMask)` Vstupy do funkce
 - `XXX (XXX)` - `XXX`.

Výstupy z funkce

- `XXX (XXX)` - `XXX`.

Zde vložit popis

Zde vložit obrázky - příklad nalezených bodů

5.5.3 Získání matematických příznaků látky ze vzdáleností bodů v čase

- `points = getDepthValueOfPoints(frontOfDepth, pointPattern, area)` Vstupy do funkce
 - `XXX (XXX)` - `XXX`.

Výstupy z funkce

- `XXX (XXX)` - `XXX`.

Zde vložit popis

Zde vložit ukázkou získaných dat

Literatura

- [0] Neoral Michal: *Aktuální verze tohoto návodu* [online].
Dostupné z: [https://github.com/michalneoral/collect_data_documentation/raw/master/manual_collect_data\(cze\).pdf](https://github.com/michalneoral/collect_data_documentation/raw/master/manual_collect_data(cze).pdf)
- [1] CloPeMa: *Stránky projektu CLoPeMa* [online]. [cit. 2014-02-02].
Dostupné z: <http://clopemaweb.felk.cvut.cz/>
- [2] CloPeMa: *Wikipedie projektu CLoPeMa* [online]. [cit. 2014-02-02].
Dostupné z: <http://clopema.felk.cvut.cz/redmine/projects/clopema/wiki>
- [3] CloPeMa: *Wikipedie projektu CLoPeMa - technické záležitosti* [online]. [cit. 2014-02-02].
Dostupné z: http://clopema.felk.cvut.cz/redmine/projects/clopema/wiki/Technical_Stuff
- [4] CloPeMa: *Wikipedie projektu CLoPeMa - Balíčky projektu CloPeMa* [online]. [cit. 2014-02-02].
Dostupné z: http://clopema.felk.cvut.cz/redmine/projects/clopema/wiki/CloPeMa_Packages
- [5] CloPeMa: *Wikipedie projektu CLoPeMa - Nastavení omezení rychlosti* [online]. [cit. 2014-02-02].
Dostupné z: http://clopema.felk.cvut.cz/redmine/projects/clopema/wiki/Robot_configuration
- [6] Ben Charrow: *Toolbox pro Matlab určený ke čtení rosbag souborů* [online]. [cit. 2014-02-02].
Dostupné z: http://clopema.felk.cvut.cz/redmine/projects/clopema/wiki/Robot_configuration
https://github.com/bcharrow/matlab_rosbag
- [7] Neovision: *Industrial Vision System* [online]. [cit. 2013-11-12].
Dostupné z: <http://www.neovision.cz/cz/sols/clopema.html>
- [8] Václav Jahoda: *CloPeMa* [online]. [cit. 2013-11-27].
Technický výkres 12010_00_04_Robot.pdf - UPRAVENO
Dostupné z: clopema.felk.cvut.cz/redmine/attachments/download/121/12010_00_04_Robot.pdf
- [9] Motoman: *YASKAWA Electric Corporation* [online]. [cit. 2013-11-27].
Technický výkres robotického ramena MA1400 - UPRAVENO Dostupné z:
http://www.motoman.com.tr/index.php?eID=tx_nawsecuredl&u=0&file=uploads/tx_catalogrobot/MA1400_15.pdf&t=1385654785&hash=ba1cae139705f02080bcf1f7c60f741c8df7434c
- [10] ŠONKA, Milan, Václav HLAVÁČ. *Image processing, analysis, and machine vision: a MATLAB companion*. 3rd ed. Toronto: Thomson, 2008. s.665-666, ISBN 978-0-495-08252-1.