Универзитет „Св. Кирил и Методиј“ – Скопје, Р. Македонија  
Природно – математички факултет  
Институт за информатика

Насока: Програмско инженерство

**  
Дипломска работа**

Компјутерска визија во роботика

|  |  |
| --- | --- |
| ***Ментор:***  *Доц. Д-р Невена Ацковска* | ***Студент:***  *Кристијан Петков*  *број на индекс: 10048* |

Скопје, 2013

|  |  |
| --- | --- |
| **Ментор:** | **Доц. Д-р Невена Ацковска** Природно – математички факултет, Институт за информатика |
| **Членови на комисијата:** | **Доц. Д-р**  Природно – математички факултет, Институт за информатика  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Природно – математички факултет, Институт за информатика  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Природно – математички факултет, Институт за информатика |
| **Датум на одбрана:**  **Научна област:** | \_\_\_\_.\_\_\_\_.\_\_\_\_\_\_\_\_ год.  Информатика/Роботика – |

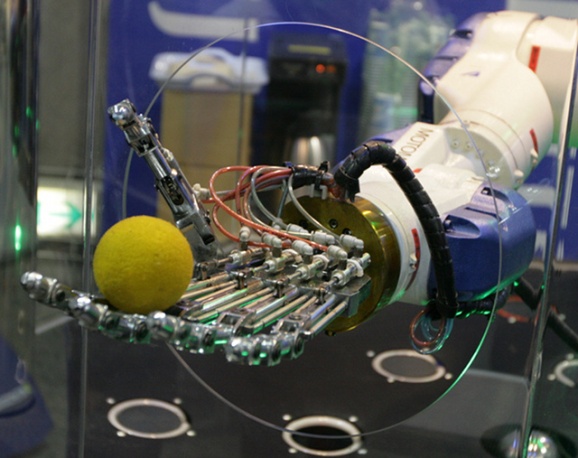
Содржина

1. **Апстракт**4
2. **Вовед**4
3. **Опис на проблемот** 5
4. **Кинематика**
   1. Frames 6
   2. Трансформации 6
      1. Координатен систем 6
      2. Операции 6
   3. Параметри на Данавит-Хатенберг7
   4. Forward kinematics 8
   5. Inverse kinematics8
5. **Препознавање на објекти**
   1. Алгоритам Background subtraction9
   2. Алгоритам Connected components10
   3. Алгоритам за Labeling10
   4. Бинарна мапа11
   5. Калибрирање на видео камера 12
6. **Дизајн на системот**
   1. Избор на платформа14
   2. Модули14
   3. Модул за кинематика15
   4. Модул за детектирање на објекти17
   5. Модул за виртуелна околина20
   6. Модул за контрола на роботска рака22
7. **Заклучок**23
8. **Подобрувања во иднина**24
9. **Користена литература**25
10. Апстракт

Визијата е екстремно важно сетило за луѓето и роботите, обезбедувајќи детални информации за околината. Робустен систем за визија треба да биде во можност да детектира објекти со голема сигурност и да обезбеди точна репрезентација на светот за процесите во системот кои се на повисоко ниво. Системот за визија мора да биде ефикасен овозможувајќи му на агентот со ограничени ресурси да одговори на промените во околината. Секоја слика добиена од дигитална камера мора да биде обработена за кратко време. Поради ова алгоритамската сложеност е ограничена, и e потребен компромис помеѓу времето на обработка и квалитетот на добиените информации. Примена во роботика: Системите за визија се главни уреди за перцепција и автономните роботи мора да бидат во можност да ги користат истите за да може самите да се лоцираат во околината и да лоцираат објекти со кои што треба да манипулираат.

1. Вовед

Повеќето копнени животни имаат способност да перцепираат објекти со различни димензии и форми а некои можат да ги препознаваат нивните карактеристики и да ги запомнат. Животните како и луѓето се во состојба да препознаваат објекти користејќи го нивното сетило за вид во околината во која се наоѓаат и успешно да манипулираат со нив користејќи ги нивните екстремитети. Инспириран од природата предизвикот за роботските системи е насочен кон копирање на вештините на биолошките организми. Предизвикот се состои во препознавање на објекти во реално време со вештачки систем за визија и манипулирање со нив. Ваквиот систем треба да биде робустен и во ограничено време со ограничени ресурси да препознае објекти и да одучи кои акции ќе ги изврши за да манипулира со објектите. Роботските системи вклучуваат сензори за вид-дигитална камера и манипулатори -роботски раце. Некои од овие роботски системи можат да бидат и мобилни, со тенденција за постојано усовршување со цел да се постигне комплетна автономност на системот.

**

*Слика 1: Малипилирање со објекти*

1. Опис на проблемот

Во работната околина на роботот може да се наоѓаат објекти со различна големина, боја и форма. Се поставува прашањето како да се препознаат објектите во работната околина на роботот, да се определи нивната релативна положба во однос на основата на роботот, да се најде нивната боја форма и ориентација и овие информации да се пренесат на процес на повисоко ниво. Овој процес во зависност од задачата треба да одлучи како ќе манипулира со дадените објекти. На пример да се сортираат објектите според дадена боја или форма. За да се реши овој проблем потребно е да се разгледаат неколку аспекти. Еве неколку од нив:

* Дали дадената точка е во работната околина на роботот
* Доколку дадената точка е на дофат како да се најдат вредности за зглобовите на роботот да ја достигне целната точка.
* Доколку се најде решение за вредностите на зглобовите дали тоа решение е во границите на роботот.
* Ако решението е во границите на роботот системот треба да провери дали даденото решение нема да доведе до судир со некој објект на сцената.

Овие аспекти се разгледани во областа на кинематика каде се имплементирани различни техники и методи за да најдат решенија со дадените ограничувања.

Сега кога имаме кинематички модел на роботот кој што овозможува придвижување на роботот кон дадената цел со дадени ограничувања се поставува проблемот како да се детектираат објектите кои се наоѓаат на сцената. Потоа информациите за нивната локација и ориентација треба да се пренесат до системот за кинематика кој е задолжен за движење на роботот.

Доколку системот за детекција открива објекти и испраќа информации за истите, а системот за кинематика соодветно наоѓа решенија за вредностите на зглобовите на роботот и успешно го придвижува, се поставува прашањето како да се симулира работата на системот за да се потврди исправното функционирање на системот во целина и воедно да се избегнат можни оштетувања на роботот при погрешно пресметани вредности.

1. Кинематика
   1. Frames

Секоја точка во тридимензионален простор е определена со координати (x, y, z). Ориентацијата на дадена точка во просторот може да се претстави со три единечни вектори: Rx(1, 0, 0), Ry(0, 1, 0) и Rz(0, 0, 1). Ако овие вектори се запишат во матрица со хомогена репрезентација тогаш секоја точка во просторот зададена со локација и ориентација може да се запише на следниов начин:

* 1. Трансформации
     1. Координатен систем

Координатниот систем е Декартов. Притоа се користи правило на десна рака, односно Z+ има ориентација нагоре, X+ кон нас, Y+ на лево. Позитивна ротација е во спротивна насока на стрелките на часовникот, додека негативната ротација е во насока на стрелките на часовникот.

* + 1. Операции

**Транслација**Транслацијата може да се изврши во било која насока, за произволна вредност. На пример транслација на фрејмот fr1 за дадени вредности (10, 12, 15) соодветно по X, Y и Z оски е пресметана на следниот начин

X =

**Ротација**

Ротацијата може да може да се изврши околу било која оска. Пример за ротација околу X оската за 90 степени.

X =

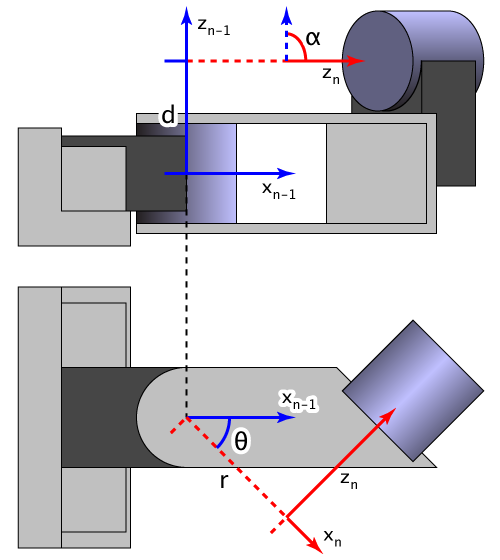
* 1. Данавит-Хатенберг параметри

Данавит-Хатенберг параметри (DH параметри) се четири параметри кои се однесуваат на конвенција за поставување референтни фрејмови на врските во кинематички синџир или роботски манипулатор. Референтните фрејмови се поставени на следниов начин :

* + 1. Z – оската е во насока на оската на зглобот
    2. X – оската е паралелна со заедничката нормала . Доколку не постои заедничка нормала d е слободен параметар.
    3. Y – оската се користи да го комплетира фрејмот, користејќи го правилото на десната рака

Трансформацијата е опишана преку четирите параметри :

1. **d** –поместување по претходната Z-оска до заедничката нормала
2. **θ** – агол околу претходна Z оска од старата/претходна X оска до новата
3. **r**– должина на заедничка нормала, доколку станува збор за ротирачки зглоб овој параметар претставува радиус на ротација
4. **α** –агол околу заедничка нормала од претходната Z оска до новата Z оска



*Слика 2: Данавит-Хатенберг параметри*

* 1. Forward kinematics

Проблемот на кинематика се однесува на употреба на кинематички равенки на роботот за да се пресмета положбата на крајната точка на роботскиот манипулатор за дадени вредности на параметрите на зглобовите. Кинематичките равенки се користат во роботика, компјутерски игри и анимација. Обратниот процес кој за дадена положба на крајната точка на манипулаторот ги пресметува параметрите на зглобовите се нарекува инверзна кинематика. Кинематичките равенки за робот се добиени со користење на крути трансформации[Z]( ротација), да го опише релативното движење дозволено на секој зглоб и посебна трансформација [X] да ги дефинира димензиите на секој линк на роботот. Резултатот е низа од крути трансформации (ротација и транслација ). Оваа низа се состои од наизменични трансформации на зглобовите и линковите од основата на манипулаторот до крајната точка:

Каде Т е трансформација која ја дефинира крајната точка на манипулаторот. Овие равенки се наречени кинематички равенки на сериски кинематички синџир.

* 1. Inverse kinematics

Тривијално е да се пресмета крајната точка на кинематички синџир за дадено множествово на агли на зглобовите. Обратниот процес не е тривијален бидејќи многу често има повеќе решенија а во некои случаи нема решенија. Треба да го решиме проблемот на инверзна кинематика во анимација и роботика, каде роботот се состои од неколку кинематички синџири и треба да ја достигне целната точка во правоаголен (Декартов) координатен систем, каде што само координатите се познати.

Постоечките алгоритми кои го решаваат проблемот на инверзна кинематика користат нумерички итеративни процедури. Друг вообичаен метод за пресметување инверзна кинематика е конструкција на матрица на Јакобиан и потоа за оваа матрица се бара инверзна матрица или се транспонира. Воедно методот на инверзна матрица на Јакобиан е искористен во овој проект за да се најадат решенија за вредностите на зглобовите.

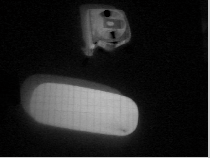
1. Препознавање на објекти
   1. Алгоритам Background subtraction

**Background subtraction** е широко користена класа техники за сегментирање на објекти од интерес на сцената. Популарноста на алгоритамот главно доаѓа од ефикасноста на пресметките, што овозможува примена во области како што се: интеракција човек - компјутер, видео надзор и набљудување на сообраќајот. Објектите на сцената се детектираат преку разликата помеѓу моменталната слика и сликата со статичка позадина:

Пример:

*Слика 3: Позадина Слика 4: Објекти на сцена*

*Слика 5: Разлика Слика 6: Бинарна мапа*

* 1. Алгоритам: Connected components

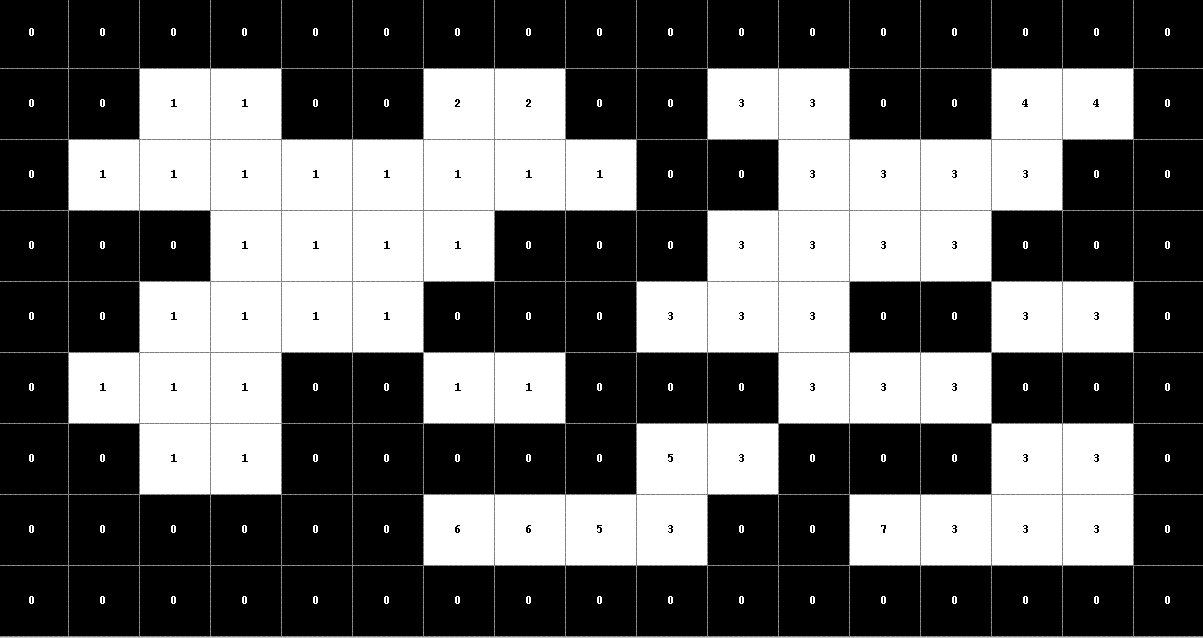
Connected components алгоритамот се користи во компјутерска визија за детектирање на поврзани региони во бинарни дигитални слики, меѓутоа можат да бидат обработени слики во боја и податоци со повисоки димензии. Кога овој алгоритам е интегриран во препознавање на објекти или во интеракција човек-коппјутер истиот може да се примени на широк опсег на информации. Генерално се користи за конструкција на региони од добиената бинарна слика. Во понатамошните чекори регионите можат да бидат филтрирани, броени и следени.



*Слика 7: Бинарна мапа*

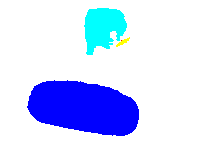
* 1. Алгоритам за ознчување (Labeling)

Кога ќе се определи поврзаноста на регионите потребно е секој регион да се означи со цел да се идентификуваат различните региони за понатамошна обработка. Означувањето на различни региони алгоритамот може да го изврши според параметрите за поврзаност. Пример поврзаност со 8 соседни точки или поврзаност со 4 точки.



*Слика 8: означени региони*

Пример, секој регион е означен со различна вредност и таа вредност е претворена во RGB за да се конструира слика која е видлива и разбирлива за човекот.



*Слика 9: Означени региони на бинарна мапа*

* 1. Бинарна мапа

Бинарна мапа или бит мапа претставува матрица со исти димензии како и сликата што се процесира. Вредностите на бит мапата како што и самото име имплицира се 0 или 1. Бинарната мапа се користи најчесто во процесот за сегментирање на региони на сликата кои што се од интерес. Исто така може да се употребат различни техники на филтрирање на регионите, според површина, форма итн. На пример бит мапа добиена со примена на алгоритамот background subtraction.



*Слика 10: Бинарна мапа*

* 1. Калибрирање на видео камера

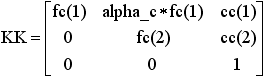
Најчесто цел за фат на објектите е центарот на самиот објект . Со комбинирање на техниките споменати погоре добиваме механизам за детектирање на објекти во работната околина на роботот. Но координатите што се добиени преку детекција на објектите се во координатен систем на сликата односно (x, y). Потребно е да се најде трансформација од координатниот систем на сликата во реалниот свет така што за секоја точка на сликата I=(x, y), би добиле точка во реалниот свет W=(x, y ,z), во координатен систем на камерата , притоа мерните единици се во милиметри. За да ја најдеме трансформацијата потребно е да извршиме постапка наречена калибрирање на камерата. Оваа постапка овозможува да ги добиеме внатрешните параметри на камерата преку кои се извршува проекцијата.

Внатрешни параметри на камерата :

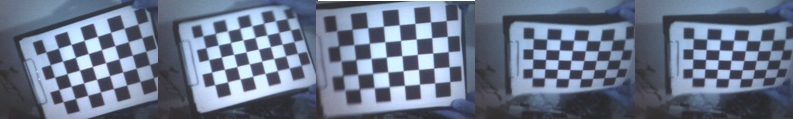
* **Фокално растојание :** Ова растојание е изразено во милиметри помеѓу леќите и сликата што се формира на сензорот, кога сликата е остро фокусирана во бесконечност. Фокалната должина го определува аголот на поглед или колку леќата гледа, со што се контролира делот од сцената што ќе биде опфатен. Фокалната должина во пиксели е зачувана во вектор 2х1 означен со **fc.**
* **Внатрешна точка (или центар на сликата):** Точка на пресек помеѓу оптичките оски и рамнината на сликата. Координатите на внатрешната точка се зачувани во 2х1 вектор означен со **сс**
* **Коефициент на накосеност:** Коефициентот на накосеност дефинира агол помеѓу x и y оските на пикселите, и е зачуван во скалар **alpha\_c.**
* **Дисторзии**: Коефициенти на дисторзија на сликата(радијални и тангентни) се зачувани во вектор 5х1 означен со **kc**

Процесот на калибрирање на камерата започнува со зачувување на индивидуални слики од испечатена шема слична на шаховска табла која се наоѓа пред камерата. Притоа шаховската табла на секоја слика треба да има различна ориентација од претходната. Осум до десет слики се доволни за да се определат внатрешните параметри на камерата. Софтверот користен за калибрирање на камерата е “[**Camera Calibration Toolbox for Matlab**](http://www.vision.caltech.edu/bouguetj/calib_doc/)”

Од добиените параметри може да се конструира матрица на проекција, со која секоја точка од сликата I=(x,y) може да се трансформира во 3Д координатен систем на камерата:



*Слика 12: Матрица на проекција*



*Слика 11: Калибрирање на камера*

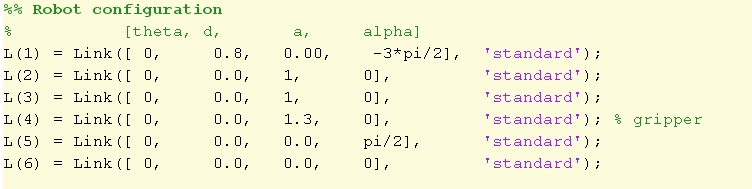
1. Дизајн на системот
   1. Избор на платформа

Платформата во која е имплементиран софтверскиот систем е MATLAB. При изборот на платформа за реализација на софтверскиот систем земени се повеќе критериуми:

* Лесен начин на изразување на математички пресметки.
* Достапност на различни алатки со цел максимално искористување на постоечки технологии.
* Еден од најважните критериуми при изборот е и можноста за дизајнирање на симулатор кој ќе овозможи симулирање на роботскиот систем во реално време со голема точност.
* Алатки кои овозможуваат обработка на слики.
* Алатки кои овозможуваат калибрирање на дигитални камери
  1. Модули

Системот се состои од посебни модули кои комуницираат помеѓу себе преку глобални променливи кои се сместени во меморијата на Матлаб. Системот се состои од следниве модули :

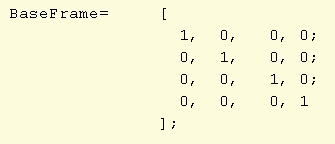
* Модул за кинематика. (Кинематички солвер кој овозможува пресметување на инверзна кинематика со неколку алгоритми)
* Модул за детектирање на објекти
* Модул за трансформација на координати од сликата во координати на роботот
* Модул за виртуелна реалност
* Модул за комуникација со роботот/ контролер на роботот
* Модул за графички интерфејс со корисникот
  1. Модул за кинематика

Прв чекор во кинематичките пресметки е опис на конфигурацијата на роботот. Најчест метод за опишување на конфигурацијата се параметрите на Данавит – Хатенберг кои на едноставен начин овозможуваат дефинирање на кинематичките линкови на роботот.

*Слика 13: Конфигурација на робот со DH параметри*

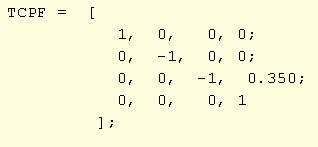
Вредностите за должините на линковите на роботот се скалирани со што се олеснува репрезентацијата на роботот. Роботот има 4 зглобови, што значи четири степени на слобода, но како што може да се забележи од сликата погоре во конфигурацијата на роботот има шест степени на слобода од кои последните два се дополнителни и се наоѓаат на крајната точка на роботот. Причината за додавање на дополнителни степени на слобода е да се овозможи достапност на секоја локација со дадена ориентација која се наоѓа во работната површина на роботот.

Покрај конфигурацијата на роботот потребно е да го дефинираме фрејмот на основата(Robot Base Frame) . Овој фрејм ја дефинира локацијата на роботот во тридимензионален простор. Притоа сите локации кои се дефинирани во работната околина на роботот се релативни во однос на фрејмот на основата. Во нашиот случај роботот е поставен во центарот на координатниот систем.



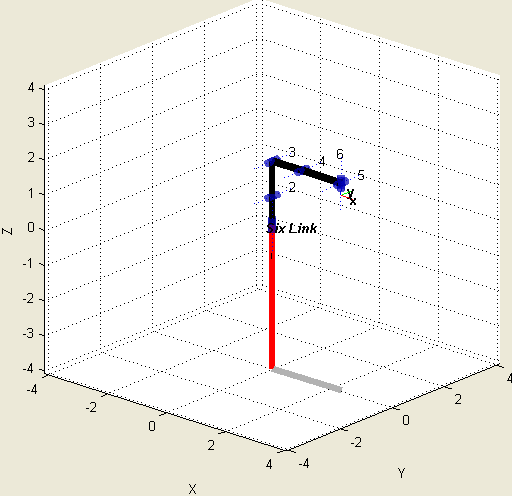
*Слика 14: Фрејм на основата на роботот*

Сега кога ја имаме дефинирано конфигурацијата на роботот следен чекор е дефинирање на TCPF(Tool Point Center Frame). Обично TCPF е поместување од крајната точка на роботот. Со ова поместување е претставена крајната точка на алатката што е поставена на роботот. Во нашиот случај алатката е „клешта“ монтирана на роботската рака.



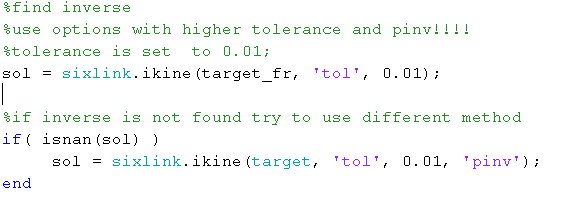
*Слика 15: Конфигурација на TCPF*

Многу често за дадена локација во просторот постојат повеќе од едно решение за вредностите на зглобовите на роботот. Обично локацијата е достигната со лактот нагоре, или со лактот надолу. Во нашиов случај единствено решение е конфигурација на роботот со лактот нагоре, бидејќи доколку го избереме решението со лактот надолу, вредностите ќе ги надминат границите на зглобовите на роботот или ќе доведат до судирање на роботот со подот.



*Слика 16: Кинематички модел на роботот*

ММодулот за кинематика наоѓа решенија за вредностите на зглобовите за дадена локација. Притоа ја користи методата на транспонирана матрица на Јакобиан. Алгоритамот користи итеративен метод за наоѓање на решенијата. Доколку овој алгоритам не може да најде решенија за дадената локација тогаш се применува псевдо-инверзен метод наместо транспонирана матрица на Јакобиан.



*Слика 17: Употреба на методи за инверзна кинематика*

* 1. Модул за детектирање на објекти

Модулот за детектирање на објекти овозможува детекција на објекти кои се наоѓаат во работната околина на роботот. Со примена на алгоритамот за Background Subtraction, модулот е во можност да конструира бинарна мапа. По употребата на овој алгоритам во модулот за детектирање на објекти се употребени неколку додатни техники кои овозможуваат попрецизно и посигурно детектирање на објектите. Постапката се состои од неколку чекори:

* Сликите во боја добиени од дигиталната камера се претвораат во gray-scale (црно бели слики). Оваа постапка се применува со цел да се намалат пресметките и да се зголеми брзината на обработка.
* Кога референтната слика и сликата со објекти се претворени во gray-scale слики, се применува алгоритамот Background Subtraction, од што се добива слика во која се истакнати разликите на секој пиксел од сликата



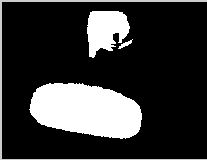
*Слика 18: Примена на Background Subtraction алгоритмот*

* Следен чекор е отстранување на “шумот” од добиената слика. Со примена на **wiener2** филтерот кој е достапен во алатките на Матлаб за обработка на слики со интензитет се отстранува шумот кој е добиен од камерата. Овој метод користи статистички податоци од соседните пиксели за да го отстрани шумот. Во нашиот случај е земена околина од 5х5 пиксели.
* Сега може да се изгради бинарна мапа со примена на гранична вредност. Односно сите пиксели од сликата кои имаат вредност поголема од граничната се обележани со 1 додека пикселите чиј интензитет е под граничната вредност се обележани со 0. Резултатот од оваа постапка е прикажан на следната слика:



*Слика 19: Бинарна мапа*

* Како што може да се забележи од сликата погоре во горниот објект се наоѓаат празнини кои се добиени поради различно осветлување. За да се овозможи детектирање на објектите со поголема сигурност се користи метод за исполнување на вакви празнини. По примената на ова метода се добива бинарна мапа во која се отстранети празнините во регионот.



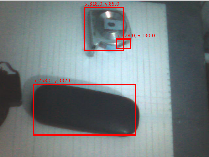
*Слика 20: Бинарна мапа со исполнети региони*

* Следен чекор е да се обележат регионите со примена на алгоритамот Connected Components:



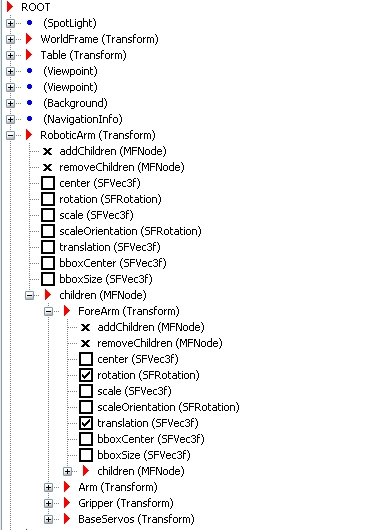
*Слика 21: Обележани региони*

* Како што може да се забележи на сликата мал сегмент од горниот објектот е отцепен. Ова е исто така резултат од различното осветлување на самиот објект како и својствата на материјалот. Идеално, за да се избегне фрагментација на објектот постои техника која врз основа на оддалеченоста на регионите и нивната големина одлучува дали два блиски региони ќе ги поврзе.
* Сега кога имаме мапа со означени региони може да примениме уште еден алгоритам кој ќе ги отстрани сите региони кои се помали од зададена вредност со цел да се добие почиста слика. Во нашиот случај сите региони со површина помала од 200 пиксели се отстранети.
* Бидејќи од интерес се објекти кои се наоѓаат во центарот на сликата може да примениме уште една техника која отстранува региони кои се наоѓаат по рабовите на сликата.
* Последен чекор е да го идентификуваме центарот на секој регион, неговите димензии како и површината.
* Откако овие информации ни се познати околу регионите може да исцртаме правоаголници кои ќе ги обележат регионите.



*Слика 22: Детектирани објекти*

* Сега информациите за центарот на објектите може да се пренесат до наредниот модул за понатамошна обработка
  1. Модул за виртуелна околина



*Слика 23: Хиерархија на компоненти*

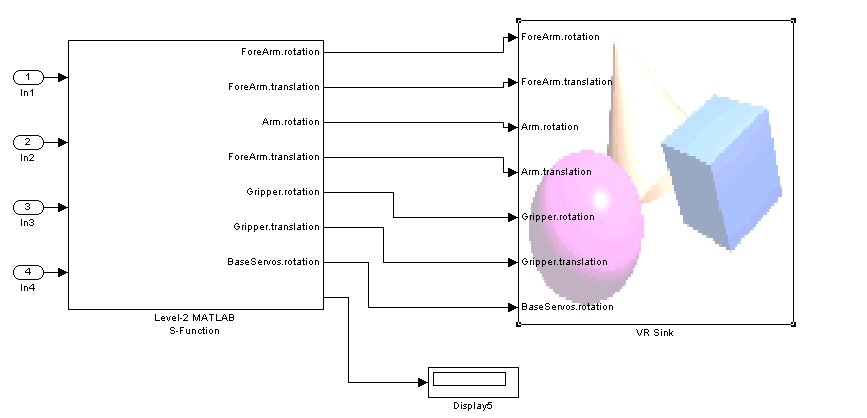
*во трансформација*

Модулот за виртуелна околина овозможува симулирање на работата на роботот во реално време. Симулацијата овозможува да ја провериме работата на секој модул индивидуално како и резултатите од нивната интеграција. Модулот се состои од подсистем кој ги извршува пресметките за анимација на роботските компоненти. Влез на системот се вредности за зглобовите на роботот. Подсистемот соодветно ја пресметува транслацијата и ротацијата за секоја индивидуална компонента.

Технологијата која се користи за создавање на виртуелна околина е **VRML (Virtual Reality Modeling Language)** . VRML е стандарден формат за претставување на 3 димензионална интерактивна вектор графика. Овој стандард е дизајниран според World Wide Web односно Интернет. Бидејќи станува збор за јазик, рабовите и линиите на тридимензионален полигон можат да бидат специфицирани заедно со бојата на површината, текстури, како и осветленост на објектите итн. Оваа технологија Матлаб ја интегрира во Simulink овозможувајќи да се претстават некои системи во виртуелна реалност и извршувањето на симулацијата да биде прикажана во виртуелен свет. Во VRML постои хиерархија на компоненти со што се овозможува групирање. Соодветните компоненти се сместени во трансформација, притоа компонентите се деца на трансформацијата.

Оваа трансформација ни овозможува потоа да ги анимираме компонентите кои се наоѓаат во нејзината хиерархија преку транслација, ротација, скалирање итн. Во нашиов случај од интерес се „крути“ трансформации односно ротација и трансформација.

Simulink овозможува да бидат селектирани соодветните операции. Селектираните операции се појавуваат во блокот како влезни порти.

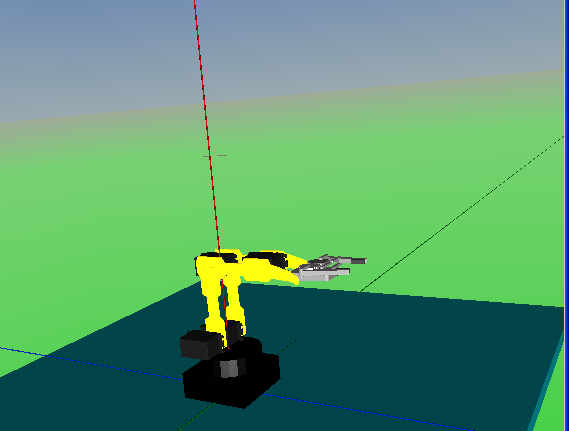


*Слика 24: Компоненти на симулаторот*

Роботската рака се состои од следниве компоненти со соодветните операции кои се прикажани на сликата погоре.

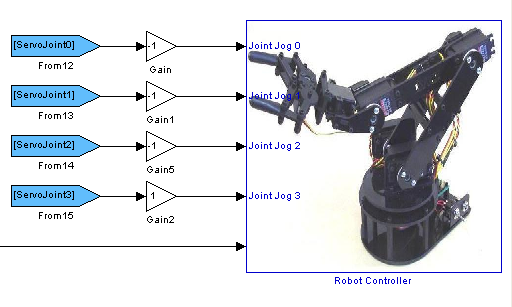
* Серво мотори на основата
* Подлактица
* Надлактица
* Клешта

Соодветните операции (ротација и транслација) се пресметани во под системот кој за дадени вредности на зглобовите(влезови) на роботот пресметува каде ќе биде позицијата во виртуелниот свет на компонентите на роботската рака.



*Слика 25: Тридимензионална виртуелна околина на роботот*

* 1. Модул за контрола на роботска рака



*Слика 26: Модул за контрола на роботска рака*

Овој модул овозможува испраќање на контролни сигнали до роботскиот конролер. Притоа потребно е вредностите кои се испраќаат да бидат соодветно изразени во формат познат на контролерот. Овој модул исто така го определува типот и брзината на поврзувањето со контролерот. Во нашиот случај станува збор за **Mini SSC II Serial Servo Controller**. Што значи дека поврзувањето е исклучиво преку сериска порта. Овој контролер го користи познатиот индустриски стандард RS232. Комуникацијата помеѓу модулот и конролерот е реализирана преку испраќање на контролна низа од бајти:

|  |  |  |
| --- | --- | --- |
| Byte 1 | Byte 2 | Byte 3 |
| [Синхронизација (255)] | [Серво # (0-254) | [Позиција (0-254)] |

Како што може да се забележи контролните вредности за позицијата на серво моторите изнесуваат од 0 до 254. Ваквиот стандард наметнува потреба за претворање на вредностите на зглобовите на роботот од агли во вредности за позиција. За оваа цел се користи формулата :

*Servo Position = (127/45)\*angle + 127*

На овој начин се овозможува внесување на позитивни и негативни вредности за агли во опсег од [-45, 45] степени. Овозможувајќи на модулот за кинематика директно да ги пренесе вредностите на решенијата од инверзната кинематика.

1. Заклучок

Веќе подолго време технологијата за обработка на дигитални слики како и алгоритмите за препознавање на објекти се достапни. Исто така роботските системи се достапни подолго време. Роботските манипулатори се применуваат во индустријата повеќе од две децении. Но интеграцијата на овие две технологии започна неодамна. Постојат повеќе причини за ова доцнење, главните причини се ограничувањата од самите хардверски уреди, како на пример недоволно брзи процесори кои ќе овозможат обработка на дигитални фотографии и пресметка на кинематика во реално време. Друго ограничување е самиот квалитет на информациите што се добиваат од дигиталните фотографии. Но како што технологијата еволуира и достапни се се’ побрзи процесори,интеграцијата на овие технологии веќе станува реалност. Во индустријата се прават напори да се интегрираат стерео камери со индустриски роботи.

Реализацијата на овој проект е само концепт за интеграција на неколку постоечки технологии и овозможување на овие технологии да „комуницират“ меѓусебно со цел да се изгради робустен систем кој е во можност да биде во интеракција со надворешната околина.

После десетина успешно извршени експерименти, системот беше во можност со голема точност да ги препознае објектите кои се наоѓаат во работната околина и успешно да ја придвижи роботската рака со цел да се изврши некаква операција врз објектите кои се наоѓаат во околината. Но како и кај секој систем постојат недостатоци и ограничувања. Некои од нив се проблемот со осветлувањето на објектите во процесот на препознавање. Друг реален проблем е кога веќе објектот е препознаен како да се најде точка на самиот објект која ќе претставува идеална локација за фаќање. Овие проблеми ќе бидат опфатени во иднина.

1. Подобрувања во иднина

Подобрувањето на системот може да се изврши во неколку модули со што би се постигнала поголема флексибилност на самиот систем во целина.

* Во пресметките за инверзната кинематика да се вклучат и граници на аглите бидејќи постојат физички ограничувања на самата роботска рака.
* Некои локации во зависност од конфигурацијата на роботот може да бидат достапни со неколку пози, односно да постојат неколку решенија. Модулот за инверзна кинематика треба да ги враќа сите можни решенија за дадена локација.
* Можност за детектирање на судир со објекти во околината и наоѓање на решенија со кои ќе избегне судир.
* Модулот за препознавање на објекти да биде во можност да препознава карактеристики на објекти, како што се рабови со што би се избегнал проблемот при сегментирање кој настанува поради осветлувањето на објектите како и можност за отстранување на сенки.

1. Користена литература

[1] K. C. Gupta and K. Kazerounian, “Improved numerical solutions of inverse kinematics of robots,” in Proceedings of the 1985 IEEE International Conference on Robotics and Automation,

vol. 2, March 1985, pp. 743–748.

[2] A. Goldenberg, B. Benhabib, and R. Fenton, “A complete generalized solution to the inverse kinematics of robots,” IEEE Journal of Robotics and Automation, vol. 1, pp. 14–20, March 1985.

[3] V. J. Lumelsky, “Iterative coordinate transformation procedure for one class of robots,”

IEEE Transactions on Systems Man and Cybernetics,

vol. 14, pp. 500–505, Jun. 1984.

[4] V. D. Tourassis and J. Ang, M.H., “A modular architecture for inverse robot kinematics,”

IEEE Transactions on Robotics and Automation,

vol. 5, pp. 555–568, October 1989.

[5] M. Fedor, “Application of inverse kinematics for skeleton manipulation in real-time,” in SCCG ’03: Proceedings of the 19th spring conference on Computer graphics. New York,

NY, USA: ACM Press, 2003, pp. 203–212.

[6] P.I. Corke, "A Robotics Toolbox for MATLAB"

[7] <http://www.petercorke.com/RVC/>

[8] <http://en.wikipedia.org>

[9] Camera Calibration Toolbox for Matlab® <http://www.vision.caltech.edu/bouguetj/calib_doc/>