

Разпознаване на човешки активности на база 3D реконструкция на скелета

<http://kinect-human-action-recognition.googlecode.com/>

Автор: Любомир Янчев;
11 клас, ЧНГ „Ерих Кестнер“;
София;
yanchev.lyubomir@gmail.com

Ръководител: Светослав Колев;
University of Washington;
swetko@cs.washington.edu

Съдържание

Абстракт	2
Въведение	2
Предишна работа	3
Методи.....	3
Ъгли и ротация на ставите	3
Представяне на ъглите чрез двуизмерни проекции координатите	3
Представяне на ротацията на ставите чрез геометрични кватерниони.....	4
Заклучение	4
Селекция на „най-информативни“ стави	4
Сравняване на движения.....	6
Dynamic time warping (DWT)	6
Разпознаване на активности в реално време	8
Използване на „прозорец“	8
Използване на маркери за начало и край на движението	8
Заклучение	8
Имплементация	9
Използвани технологии	9
Kinect SDK.....	9
C# & .NET	9
Възможни приложения.....	9
Допълнителни имплементации и тестове	10
Minimal Variance Matching (MVM).....	10
Еластично сравняване на движения	10
Извършени тестове.....	11
Заклучение	12
Използвана литература	12

Абстракт

Наскоро нашумялата тема за разпознаване на човешки действия се радва на все повече популярност. Целта на настоящата разработка е да представи качествено ново решение на посочения проблем.

В предложеното решение са използвани многоизмерни статистически алгоритми, алгоритми за филтриране на данни, алгоритми за оценка и други. Триизмерната информация за скелета се извлича от Kinect сензор. Чрез извършването на малка модификация, предложените алгоритми биха могли да се използват за разпознаване и на друг вид зависимости.

В процес на развитие е сравнение между разработения алгоритъм, традиционно използваните и някои нестандартни подходи, описани в секцията „Допълнителни имплементации и тестове“

Въведение

Разпознаването на човешки действия остава една от най-значимите сфери на изследване за компютърно зрение. Разработването на алгоритми за целта е стимулирано от многото потенциални приложения на подобна технология – от автоматизирано видео наблюдение, през базирано на съдържанието видео, управление на компютри до вграждането му в „умни домове“. Подходът може да се използва и за наскоро нашумялото предопределяне на болести на база промени в движенията на пациента – често преди инфаркт се забелязва леко завлачване на левия крак.

В миналото екстракцията на точна информация за скелета от монокулярни камери бе труден за реализация и несигурен подход, а ефективни алгоритми използващи друг вид сензори, не работеха на нужното ниво. Специализираните системи за следене на предмети, базирани на поставяне на маркери върху повърхността на обектите, предоставят изключително точни данни, но за сметка на това са приложими единствено в студийна среда, което ограничава приложението им. Изключение правят някои системи, които главно се използват за мониторинг на пациенти и други. (1) Въпреки това, те често се оказват неудобни, поради необходимостта от слагане на физически сензори по тялото на потребителя.

Поради тази причина се налага обществото от учени да се средоточи върху подходи, които не изискват поставяне на допълнителни устройства и сензори върху ползвателите. Въпреки че за момента съществуват алгоритми, използващи монокулярни камери, разпознаващи хора и предоставящи информация за скелета им, тенденцията показва, че комбинираният подход – обикновена камера и дълбочинен сензор, дава много по-точни резултати, дори конкурентни на методите, използващи маркери.

След щателно проучване на достъпните технологии установихме, че в Kinect алгоритъмът за позициониране на ставите в триизмерното пространство е най-ефективно имплементиран. Поради тази причина и качеството на разработката, взехме решението да използваме този сензор.

В раздел „Допълнителни имплементации и тестове“ са приложени тестове с други алгоритми, освен основния за тази разработка.

Преидшна работа

Множество решения на проблема, използващи различни източници на данни, са предложени през изминалите години. Информация за позата е най-често извличана от системи с маркери (1), монокулярни камери (2) (3) (4) (5), силуетни образи (6) и алгоритми, използващи дълбочинна картина (7). След цялостно проучване, не успяхме да намерим публикувана разработка, използваща Kinect сензор.

На разположение са също и алгоритми, способни да заменят Kinect сензора. Такива са например базираните на TOF камери (8), монокулярни (9) (10) (11) (12), дълбочинни (13) (14) и други (15), които бяха отхвърлени заради качеството на Kinect сензора.

Методи

Тази секция цели да изясни подробно методите използвани за реализация на алгоритъма.

Ъгли и ротация на ставите

След анализ на данните, които бихме могли да извлечем от използвания сензор, взехме решение, че най-добрият подход към проблема би бил да използваме ротацията на ставите в пространството, тъй като тя еднозначно дава информация за движението. (16)

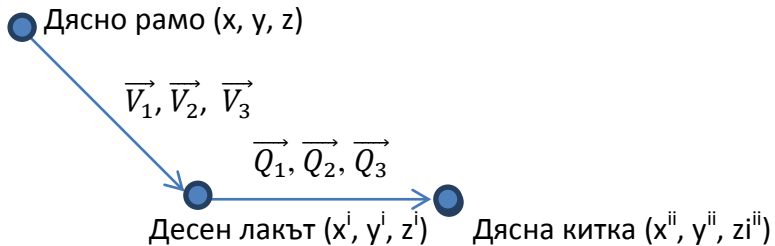
Друг плюс на използването на ротациите на ставите за първостепенна информация за разпознаване на дейностите е, че не се обвързваме с точни координати, а от там – и с точни дължини на костите на скелета.

Разработени са два подхода на представяне на ъглите:

Представяне на ъглите чрез двуизмерни проекции координатите

Макар и не реална, този метод дава еднозначна информация за ъгъла на ставата.

Същността му се състои в това да групираме двуизмерните проекции на триизмерните вектори по двойки и да намерим 3 числа, които представят ъгъла на ставата. Нека представим двуизмерните вектори $\vec{V}_1(x-x^i, y-y^i)$, $\vec{V}_2(x-x^i, z-z^i)$, $\vec{V}_3(y-y^i, z-z^i)$ и векторите $\vec{Q}_1(x^i-x^{ii}, y^i-y^{ii})$, $\vec{Q}_2(x^i-x^{ii}, z^i-z^{ii})$, $\vec{Q}_3(y^i-y^{ii}, z^i-z^{ii})$. Така, ако намерим двуизмерните ъгли между \vec{V}_1 и \vec{Q}_1 , \vec{V}_2 и \vec{Q}_2 и \vec{V}_3 и \vec{Q}_3 , то ще получим ъглите α , β и γ , чрез които после по-късно бихме могли да сравним дадения ъгъл с друг, но не бихме могли да реконструираме точните променливите, представящи векторите на ставите.



Проблем, възникващ при използването на този подход, е създадената зависимост на ориентацията на потребителя спрямо сензора – методът дава различни данни, когато потребителят е лицево ориентиран спрямо сензора, и когато е ориентиран например на 45° . Това би могло да се избегне, като се пресметне разликата между едната и другата ориентация, което дефинираме като „големина на грешка“ – колкото по-голяма е „големината на грешката“, толкова повече се различават двете ориентации на потребителя.

Можем да пресметнем големината на грешката, като намерим разликата в ориентацията вектора с начало координатите на HipCenter и край координатите на произволно HipLeft или HipRight. Така намираме еднозначно разликата в ротацията на потребителя спрямо сензора.

Въпреки интуитивността на този подход, намирането на точна зависимост на големината на грешката и пресметнатите двуизмерни ъгли, е трудоемък процес, който често дава отклонение, което е критично при използвания алгоритъм, който изисква точна и еднозначна информация за ставите.

Представяне на ротацията на ставите чрез геометрични кватерниони

Чрез прилагането на този метод бихме могли да представим ротацията и позицията на ставата чрез геометричен кватернион, който ни служи, както да сравняваме приликата между ротациите, така и да реконструираме позицията и ротацията на ставата.

Под предвид е взето, че системата на кватерниона дава както положителен, така и отрицателен резултат.

Заклучение

Имплементация на двата представени метода показва, че и двата дават добри резултати в контекста на алгоритъма. За удобство в сегашната си версия алгоритъмът използва втория метод, поради възможността от неточност при използването на първия. След обстойно тестване и с двата метода за представяне на ротацията на ставите, резултатите еднозначно показаха, че при използване на втория метод, разпознаването е по-успешно. Въпреки това, качеството на разработката и архитектурата ѝ позволяват лесното имплементиране на различни методи за в бъдеще.

Селекция на „най-информативни“ стави

Минали проучвания показват, че за разпознаване на движения на база ротацията на ставите, не е нужно да се разглеждат всички стави на скелета (16) (17).

Достатъчно е да следим единствено ставите, които имат най-голяма промяна в ротацията си по време на движението. Прост пример би било махането с ръка. Няма смисъл да взимаме предвид ъгъла на левия глезен, за да разпознаем това движение, тъй като ставите, които определят движението са рамото и лакътят. Подкрепящи това твърдение доказателства могат да бъдат намерени на (17) и на фигура 1.

Фигура 1:

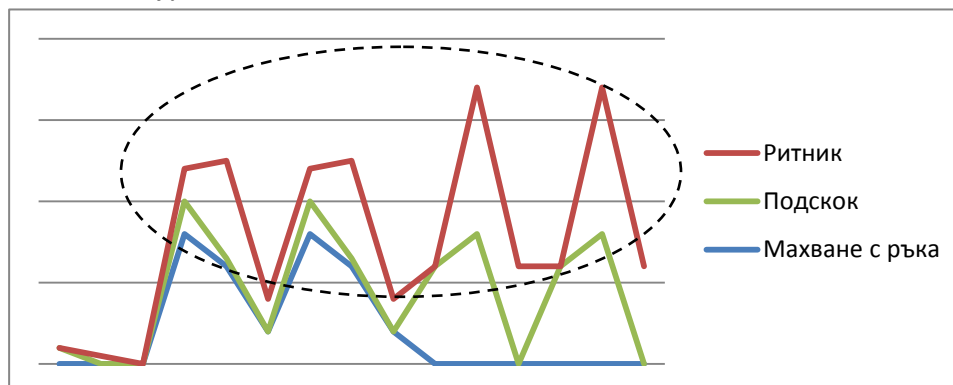


Най-активните стави се маркират при предварително записаните движения. На записите се прави анализ, който определя степента на активност по формулата.

$$\sum_{i=1}^n \|f(i+1) - f(i)\|$$

В последствие се селектират единствено „топ-ставите“ или тези, които от които се определя движението. На фигура 2 можете да видите графиката на най-активните стави и точната селекция на най-активните такива.

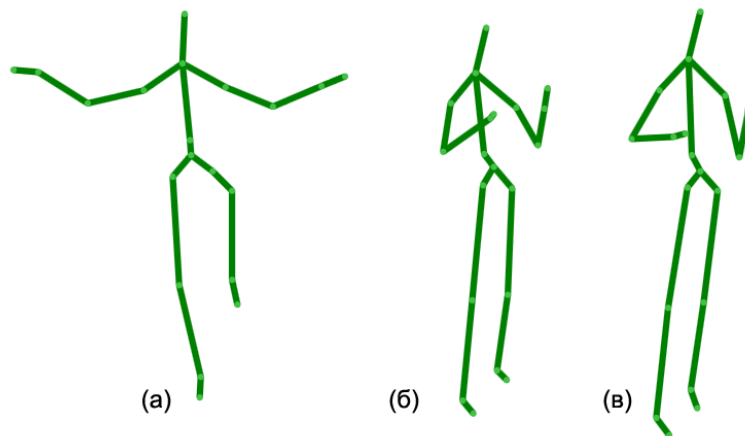
Фигура 2:



Сравняване на движения

За да решим глобалния проблем за разпознаване на активности в реално време, първо трябва да решим подпроблема за сравняване на два скелета. Прост пример виждаме във фигура 3а, 3б и 3в. Лесно можем да определим, че фигура 3б прилича много повече на фигура 3в, отколкото на 3а.

Фигура 3:



За да сравним скелетите, които ще означим с A и B , трябва да сравним ротацията на ставите им. В минали версии на разработката, в които се използваша двумерните проекции на координатите на ставите, представяхме трите проекции като координати на точка в триизмерното пространство и изчислявахме евклидовото пространство между точката от първата и втората поза. Това ни предоставяше еднозначен начин за определяне на приликата между двете стави – колкото по-малко е евклидовото разстояние, толкова повече си приличат двете стави по ротация.

В настоящата имплементация на разработката се използва методът за представяне на ротацията на ставите чрез геометрични кватерниони (R, S), които сравняваме като използваме следната формула:

$$\cos \alpha = R_x * S_x + R_y * S_y + R_z * S_z + R_w * S_w$$

Колкото по-голям е полученият косинус, толкова повече се приближават двата кватерниона.

Друг метод за сравнение на ротацията на ставите чрез кватерниони (R, S) е да използваме следната формула:

$$Difference = (R_x - S_x)^2 + (R_y - S_y)^2 + (R_z - S_z)^2 + (R_w - S_w)^2$$

Dynamic time warping (DWT)

Поради липсата на български превод на понятието, в настоящата разработка ще го наричаме „Динамично изкривяване на времето“ (ДИВ). Този

метод служи за измерване на приликата между две редици, които могат да варират спрямо време и скорост. (18) (19) (20)

Чрез прилагането на ДИВ бихме могли да намерим приликата между две активности, дори и ако те са извършени с различни скорости. Например, ако в единия случай потребителят извършва активността „удар с ръка“ с една скорост, а в другия случай я извършва по-бързо, то без да се затрудним бихме могли да открием, че в същността си двете активности представляват едно и също.

Ако не вземем предвид нуждата да разпознаваме активностите в реално време, то бихме могли да дадем следния пример. Записът на активността А има продължителност 50 фрейма, а записът на активността Б има продължителност 100 фрейма. Очевадно скоростта извършената активност Б е два пъти по-бърза, отколкото скоростта на активността А. Ако приложим линейна проверка (кадър по кадър), резултатът ще бъде грешен – приликата между двата записа би била оценена грешно, което би повлияло на вземането на крайното решение.

В следващия раздел ще разгледаме случая, в който динамичното изкривяване на времето трябва да се приложи в реално време.

Псевдо код на ДИВ в контекста на сравняване на два записа на активности:

```
int CompareActivity(Skelton[] s[1..n], Skeleton t[1..m]) {  
    declare int DynamicTimeWarping[0..n, 0..m]  
    declare int similarity  
  
    for i := 1 to m{  
        DynamicTimeWarping[0, i] := infinity  
    }  
    for i := 1 to n{  
        DynamicTimeWarping[i, 0] := infinity  
    }  
    DynamicTimeWarping[0, 0] := 0  
  
    for i := 1 to n{  
        for j := 1 to m{  
            similarity:= d(s[i], t[j])  
            DynamicTimeWarping[i, j] := similarity + minimum(DynamicTimeWarping[i-1, j ],  
                                                             DynamicTimeWarping[i , j-1],  
                                                             DynamicTimeWarping[i-1, j-1])  
        }  
    }  
    return DynamicTimeWarping[n, m]  
}
```

Поради скоростта $O(n*m)$ на този алгоритъм е възможно в бъдещото развитие на „Разпознаване на човешки активности на база 3D реконструкция на скелета“, да бъде имплементиран FastDTW (21), рекурсивен вариант на

динамичното изкривяване на времето, който работи със значително по-голяма скорост.

Разпознаване на активности в реално време

Подходът, нужен за разпознаване на активности в реално време, силно се различава от този, използван за разпознаване на активности в записи, тъй като ДИВ работи с две предварително зададени редици.

За момента предлагаме две решения на проблема.

Използване на „прозорец“

Дефинираме „прозорец“ като последователна поредица от кадри, чието начало е най-новият кадър, а дължината бележим с n . За удобство ще бележим прозореца с W .

Ако на всеки кадър прилагаме ДИВ с параметри W и всеки един от записите в базата данни, то ще постигнем теоритична сложност: $30 \text{ движения} * 30 \text{ записи} * \text{средно } 100 \text{ кадъра} * n * 60 \text{ (операции, нужни за сравняване на 2 скелета)} = 540000 * n$. От тук разбираме, че най-голямата възможна стойност на n би била 100 кадъра, за да бъде разпознаването в реално време. Ако при имплементацията се използва и процесорът на видео картата, дължината на прозореца би могла да достигне до средно 130 кадъра.

Един от вариантите, който предстои да бъде тестван, е да се дават по-малко записи на възможните движения или пък на база всички записи, да се направи „осреднен“ вариант на всички тях.

Използването на „прозорец“ би могло ни даде добри резултати, но би изисквало голяма хардуерна мощ.

Използване на маркери за начало и край на движението

Друг възможен подход за решение на проблема, е поставянето на „маркери“, които да означават началото и края на движението, за да може после кадрите между тези маркери да бъдат подадени на ДИВ алгоритъма, който да оцени до колко извършеното движение прилича на някое от записите.

Поставянето на маркери се извършва на база колко скелетът от сегашния кадър прилича на скелет от някой от началните или крайните кадри на записаните движения.

Заклучение

Разпознаването на активности в реално време е сложен проблем с много решения. В настоящата работа са предложени два подхода, чийто недостатък би бил единствено изискването на повече хардуерна мощ от среднестатистическата за домашен компютър.

Начин за оптимизация би бил използването на FastDTW, който предстои да бъде имплементиран и тестван за нуждите на проекта, както и имплементирането на настоящия алгоритъм по подходящ начин, за да използва и процесора на графичната карта.

В момента авторът работи върху сравняването на ДИВ с други алгоритми с цел намиране на най-подходящия. В процес на имплементация е и решение на проблема с използването на скритите модели на Марков. (22)

Имплементация

Най-новото работно копие на имплементацията може да бъде намерено в хранилището на проекта на адрес:

<https://code.google.com/p/kinect-human-action-recognition/>

Използвани технологии

Kinect SDK

Kinect SDK 1.5.1 е използван за получаване на данни от Kinect сензора. Заедно с Kinect SDK е използвано също и Kinect Studio, което позволява възпроизвеждането на записи в апликацията.

C# & .NET

Имплементацията на алгоритъм от този сорт се нуждае от бърз достъп до данни, добре организирани чрез обектно-ориентиран подход на програмиране. ООП също предоставя модулارността и сигурността на имплементацията. .NET и C# предоставят лесен достъп до данните от Kinect сензора.

Възможни приложения

Прилагането на алгоритъма в реални условия би спомогнало за подобрението на качеството на услугите в широк спектър от сфери. От видеонаблюдение в магазини, където се следи за „съмнителни“ активности – например прибиране на нещо в джоба, видеонаблюдение на паркинги, където се следи как точно се отваря врата – с ключ или по друг начин и т.н. Друго приложение би било интелигентна класификация на филми на база съдържанието им. Особен акцент трябва да се сложи върху системите за наблюдение на пациенти, които могат да следят за най-различни активности – от сигнал за викане на сестра, до вземане на хапчета, преминаване от стая в стая, степен на активност и т.н.

Друго приложение, към което се стреми голяма част от общността на учените в момента, е изграждането на цялостен NUI интерфейс или начин за влизане във взаимодействие с компютъра по естествени пътища – не чрез мишка и клавиатура, а чрез говор, действия и други. Чрез разработения алгоритъм лесно би могло да се разработи приложение, което бива управлявано единствено чрез човешки действия.

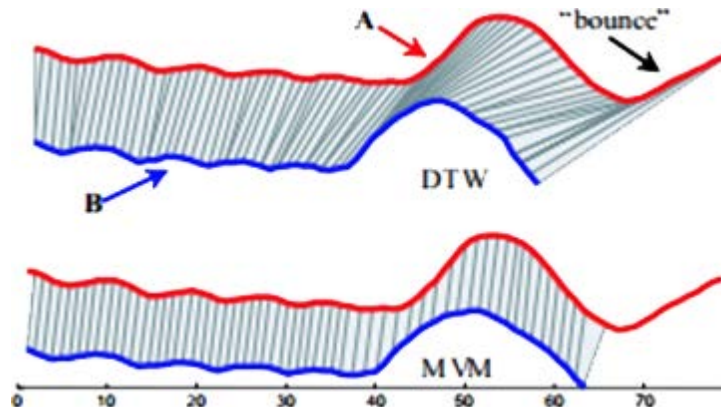
Системите за „умни домове“ все повече навлизат в нашето ежедневие. Чрез използването на предложения алгоритъм лесно може да се разширят възможностите им, като по този начин улеснят още повече ползвателите им.

Допълнителни имплементации и тестове

Minimal Variance Matching (MVM)

В изследването на учени от Temple university, Филаделфия е предложен подход за намиране на приликата между две различни по дължина редици. (23)

Фигура 4:



Подходът разчита на генерирането на матрица на разликата на всеки два елемента от двете редици, като в по-късното сравнение на редиците се взема предвид и възможността от появата на малки разлики или шум (Gaussian noise). По-късно, на база на генерираната матрица на разликите и няколко условия, бива построен асинхронен цикличен граф, в който, в последствие се прилага алгоритъм за намиране на най-кратък път, който дава еднозначно приликата между двете редици.

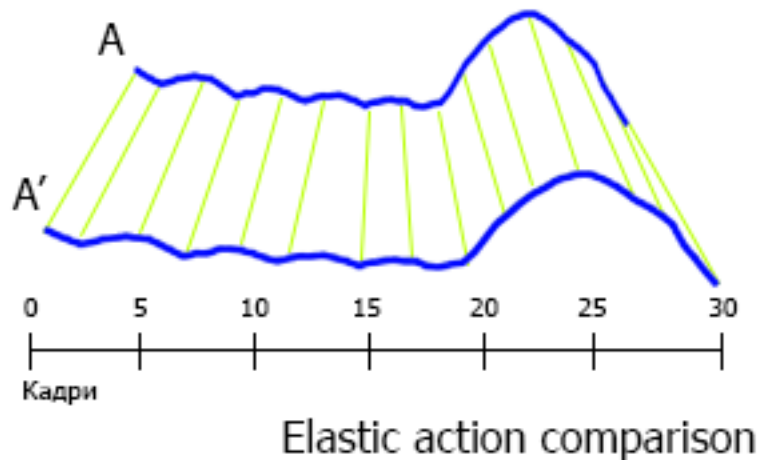
В изследването е предложен и подходящ метод за намиране на нужното скалиране и изместване на едната редица спрямо другата, за оптимални резултати.

Поради многото мнения в обществото на учените, че този алгоритъм би могъл да даде по-добри резултати в контекста на настоящата разработка и въпреки факта, че MVM допуска пропускането на елементи от едната редици при сравняването ѝ с другата, той бе имплементиран и в момента е в процес на тестване.

Еластично сравняване на движения

Еластичното сравняване на движения е подход, предложен от автора, който предстои да бъде тестван. В основата му седи хипотезата, че двете движения, които трябва да бъдат сравнени, започват с началото на движението и свършват точно с края му, поради което ще можем да сравним движения, извършени с различна скорост.

Фигура 5:



Нека означим двете движения с **A** и **B**, които са дълги съответно **m** и **n** кадъра, където $m > n$. Ако тази хипотеза е изпълнен, изчисляваме **p**, където **frameNumber** е номера на стъпката.

$$p = \frac{m}{n} * frameNumber$$

Ако означим функцията за намиране на прилика между две пози с **comp(X, Y)**, то получаваме следната формула за намиране на прилика между два записа:

$$Difference = \sum_{frameNumber=0}^n comp(A_p, B_{frameNumber})$$

Предстои да бъдат извършени тестове с алгоритъма.

Извършени тестове

След имплементирането на горепосочените алгоритми, бяха извършени 15 теста на всяко движение с всеки алгоритъм по отделно (фигура 5). Тестовите бяха проведени от автора поради невъзможността за намиране на доброволци. С увеличение на тренировъчните записи, възможността на системата да разпознава потребители, за които не е обучена, ще се увеличи.

Фигура 5:

Algorithm Action	Squat	Swipe with left hand	Swipe with right hand	Waving with left hand	Waving with right hand	Taekwondo position
Dynamic time warping	86.60%	100%	100%	91.66%	71.43%	90.91%
Elastic action comparison with freedom degree	80.00%	75.00%	81.82%	100.00%	75.57%	91.67%

От горната таблица можем да постановим, че алгоритъмът, който дава най-добри резултати, е динамичното изкривяване на времето (ДИВ). Въпреки своите недостатъци, еластичното сравняване на движения не отстъпва и след някои подобрения би могло да се справя дори по-добре от ДИВ.

Заклучение

Предложените алгоритми решават поставения проблем чрез използването на ефективен подход за подбор на данни, при който те придобиват интуитивно физическо значение. За разпознаването на действия в реално време са използвани няколко алгоритма, които предоставят релативно ефективен начин за разпознаване на еднакви движения, извършени с различни скорости. С цел по-високо ниво на качество, за набавяне на данни за триизмерния скелет на потребителя, се използва Kinect сензор. По-голяма точност на разпознаване на движенията се постига чрез филтриране на входните данни за скелета.

За в бъдеще би могло да се оптимизира времето на работа на алгоритма за динамично изкривяване на времето, както и ресурсите, които използва. Предстои да бъде направено сравнение между резултатите получени при използване на гореописаните алгоритми и най-широко използваните скрити модели на Марков. В бъдеще авторът ще направи сравнение и с получените резултатите от използването на невронни мрежи, които в днешни дни често се използват за решаване на проблеми свързани с изкуствения интелект. В момента авторът обмисля и подход, който комбинира скритите марковски модели с ДИВ с цел постигане възможно най-точно разпознаване на по-широкообхватни активности.

Друга цел за бъдещето на разработката е прилагането на алгоритъм за филтриране на „шума“ в данните, получени чрез Kinect сензора. Това рязко ще увеличи точността на всички предложени подходи в настоящата разработка и ще открие възможността за разработване на нови такива.

Благодарности

Авторът желае да изкаже благодарност на Румен Данговски, Валерия Станева, Калина Петрова и ръководителя на проекта, Светослав Колев, за ценните съвети и изказани мнения. Също така желае да изкаже благодарност и към Ученическия институт по математика и информатика (УЧИМИ) към БАН, който организира „Лятна изследователска школа“, където бе поставено началото на проекта.

Използвана литература

1. *Distributed Recognition of Human Actions Using Wearable Motion Sensor Networks.* **Allen Y. Yang, Roozbeh Jafari, S. Shankar Sastry, Ruzena Bajcsy.** 2009, Journal of Ambient Intelligence and Smart Environments.
2. *A Method For Human Action Recognition.* **Osama Masoud, Nikos Papanikolopoulos.** 2003.

3. *Qualitative Recognition of Ongoing Human Action Sequences.* **Yasuo Kuniyoshi, Hirochika Inoue.**
4. *Evaluation of local spatio-temporal features.* **Heng Wang, Muhammad Muneeb Ullah, Alexander Kläser, Ivan Laptev, Cordelia Schmid.**
5. *Human action recognition with pose similarity.* **Shiquan Wang, Kaiqi Huang, Tieniu Tan.**
6. *Expandable Data-Driven Graphical Modeling of Human Actions Based on Salient Postures.* **Wanqing Li, Zhengyou Zhang, Zicheng Liu.** 2008 r., IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, Том 18.
7. *Action Recognition Based on A Bag of 3D Points.* **Wanqing Li, Zhengyou Zhang, Zicheng Liu.**
8. *3D keypoint detectors and descriptors for 3D objects recognition with TOF camera.* **Ayet Shaiek, Fabien Moutarde.** 2011, IS&T/SPIE Electronic Imaging conference on 3D Image Processing (3DIP) and applications,.
9. *3D Reconstruction of Human Motion and Skeleton from Uncalibrated Monocular Video.* **Yen-Lin Chen, Jinxiang Chai.**
10. *Human Body Pose Recognition Using Spatio-Temporal Templates.* **M. Dimitrijevic, V. Lepetit and P. Fua.**
11. *Monocular Human Pose Estimation.* **Laxton, Benjamin.**
12. *Randomized Trees for Real-Time Keypoint Recognition.* **Vincent Lepetit, Pascal Lager, Pascal Fua.**
13. *Real-Time Human Pose Recognition in Parts from Single Depth Images.* **Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, Andrew Blake.**
14. *Real-time Upper-body Human Pose Estimation using a Depth Camera.* **Himanshu Prakash Jain, Anbumani Subramanian.**
15. *Fast Human Detection from Joint Appearance and Foreground Feature Subset Covariances.* **Jian Yao, Jean-Marc Odobez.** 10, 2011 r., CVIU, Том 115.
16. *Does Human Action Recognition Benefit from Pose Estimation?* **Angela Yao, Juergen Gall, Gabriele Fanelli, Luc Van Gool.**
17. *Sequence of the Most Informative Joints (SMIJ): A New Representation for Human Skeletal Action Recognition.* **Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, Rene Vidal and Ruzena Bajcsy.**
18. **Müller, Meinard.** *Information Retrieval for Music and Motion.* н.м. : Springer, 2007.
19. *Dynamic Time Warping Algorithm Review.* **Senin, Pavel.** 2008.
20. *Dynamic Time Warping - An intuitive way of handwriting recognition?* **Niels, Ralph.** 2004 r.
21. *FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space.* **Stan Salvador, Philip Chan.**
22. *Action Recognition by Imprecise Hidden Markov Models.* **Alessandro Antonucci, Rocco de Rosa, and Alessandro Giusti.** Milano : н.а.

23. *Partial Elastic Matching of Time Series*. **Longin Jan, Latecki, Vasileios Megalooikonomou, Qiang Wang, Rolf Lakaemper, C. A. Ratanamahatana, E. Keogh.** Philadelphia and Riverside : н.а.

24. *Expandable Data-Driven Graphical Modeling of Human Actions Based on Salient Postures*. **Wanqing Li, Zhengyou Zhang, Zicheng Liu.** 2008 г., IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, Том 18.