



**Eduardo Daniel
Ferreira Leal Pereira** **KinectMotion: activity monitoring using Kinect's skeleton
generator**

**KinectMotion: monitorização de actividades usando a
Kinect**



**Eduardo Daniel
Ferreira Leal Pereira** **KinectMotion: activity monitoring using Kinect's skeleton
generator**

KinectMotion: monitorização de actividades usando a Kinect

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática (M.I.E.C.T.), realizada sob a orientação científica do Professor Doutor José Maria Amaral Fernandes, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dedico este trabalho aos meus pais pelo seu apoio incondicional.

I dedicate this work to my parents for their unconditional support.

o júri

presidente

Prof. Maria Beatriz Alves de Sousa Santos

Professora associada com agregação do Departamento de Electrónica e Telecomunicações e Informática da Universidade de Aveiro

vogais

Prof. Miguel Tavares Coimbra

Professor auxiliar do Departamento de Ciência de Computadores da Faculdade de Ciências da Universidade do Porto

Prof. José Maria Amaral Fernandes

Professor auxiliar do Departamento de Electrónica e Telecomunicações e Informática da Universidade de Aveiro

Agradecimentos

Gostaria de agradecer em primeiro lugar ao meu professor e orientador José Maria Fernandes pelo seu apoio e disponibilidade que demonstrou ao longo da realização da minha dissertação.

Gostaria de agradecer também aos meus amigos de Aveiro que me acompanham desde novo e que a sua amizade é imprescindível.

Um obrigado aos amigos que fiz ao longo do curso e que me acompanharam ao longo deste percurso académico.

Finalmente quero agradecer aos meus pais e família mais próxima porque sem eles e sem o seu apoio certamente não teria chegado onde cheguei.

Palavras-chave

Detecção de quedas, reconhecimento de posturas, reconhecimento de atividades, monitorização domestica, segurança domestica

Resumo

Com o crescimento da população envelhecida numa escala global, todos os dias os meios de comunicação noticiam acidentes domésticos como quedas ou mesmo problemas de saúde que requerem atenção urgente. A população envelhecida a viver sozinha está inclinada para este tipo de emergências devido ao seu estatuto.

Soluções como a Kinect da Microsoft oferecem características online avançadas que permitem a detecção automática de esqueletos entre outras. Embora a Kinect seja mais usada em aplicações de jogos/lazer é viável questionar se funciona como uma solução apropriada para monitorização doméstica de baixo custo.

Nesta dissertação exploramos a utilização da extração de esqueleto automática que esta disponibiliza para suportar uma solução online de monitorização num ambiente de uma divisória que é capaz de detectar situações críticas e identificar situações típicas como quedas, sentar ou deitar. Usando técnicas simples de processamento de sinal (como filtros passa-baixo ou transformações de potencia) fomos capaz de conceber uma solução simples e fiável – KinectMotion.

KinectMotion é capaz de alertar situações críticas e detectar alterações típicas de postura numa janela temporal de 3 segundos. Na nossa avaliação sobre uma população de 6 jovens voluntários o algoritmo exibiu valores de precisão sempre superiores a 80% com um número reduzido de falsos alarmes (i.e. falsos positivos). Embora o algoritmo encontrado seja promissor necessita ser adaptado para ser aplicada à monitorização de idosos pois existem diferenças nomeadamente no desempenho motor.

Keywords

Fall detection, posture recognition, activity recognition, home-monitoring, home safety.

Abstract

With the growth of the elderly population in a global scale, every day the media reports home accidents like falls or even health problems that require urgent attention. The elderly population living alone is prone to these kinds of emergencies given their status.

Solutions such as Microsoft's Kinect's offer advance online features that enable automated skeleton extraction among other things. Although Kinect is most commonly used in gaming/leisure applications it is reasonable to ask if it provides a suitable and cost effective solution for home monitoring scenario.

In this dissertation we explore the use of Microsoft's Kinect's automated skeleton extraction to support an online monitoring solution that is able to detect critical situations in a room environment and identify typical events such as falls, sitting, lying down. Using a simple signal processing techniques (e.g. low pass filtering and power transforms) we were able to provide a simple and reliable solution – KinectMotion.

KinectMotion is able to alert to critical situations and detect typical posture changes within a time window of less than 3 seconds. In our evaluation over a population of 6 healthy young volunteers the algorithm used showed always precisions above 80%, presenting always a very low number of false alarms (i.e. false positives). Although our algorithm is promising it must be adapted to be applied to elderly monitoring as there are differences namely on motor performance.

Contents

1	Introduction.....	1
1.1.	Motivation and context	1
1.2.	Objectives.....	2
1.3.	Dissertation structure	2
2.	Posture and activity analysis.....	3
2.1.	Accelerometer Approach	3
2.2.	Video Approach	4
2.2.1.	Skeleton Approach	5
2.3.	KINECT based solutions.....	7
2.3.1.	Depth Camera Approach	7
2.4.	Summary	8
3.	KinectMotion	11
3.1.	The overall concept.....	12
3.2.	Gather skeleton data.....	13
3.3.	The Kinect Skeleton referential	14
4.	The changes detection based on Kinect extracted skeleton.....	15
4.1.	Transform Data	16
4.1.1.	Noise.....	16
4.1.2.	From pixels to meter.....	17
4.1.3.	Velocity extraction	17
4.1.4.	Relative Distance between joints.....	19
4.2.	Pattern Match through thresholds	21
4.3.	The final model	22
5.	KinectMotion algorithm validation.....	25
5.1.	Data and methods.....	25
5.2.	Subjects	25
5.3.	The procedure and the experimental protocol.....	26

5.3.1	Getting up.....	26
5.4.	Model evaluation	27
5.4.1.	Sensitivity and specificity	28
5.4.2.	Results	29
5.4.3.	Conclusions	32
6.	KinectMotion implementation.....	33
6.1.	KinectServer	34
6.2.	KinectClientAnalysisServices	34
6.3.	Communication.....	36
7.	Conclusions	37
7.1.	Future Work.....	37
8.	References	41

List of Figures

Figure 1 - Examples of Skeleton extracted from video, Kinect included.....	6
Figure 2 - Conceptual flowchart	12
Figure 3 - Skeleton Frame	13
Figure 4 - Raw Data – A fragment of a the data available in a skeleton frame with joints associated coordinates and acquisition timestamp	13
Figure 5 - Referencial transformation – X and Y coordinates transformation	14
Figure 6 - In kinect skeleton data we observed that non-moving parts presented changes. In the example the moving dots in the highlighted area in consecutive frames show changes although the subject position did not change. The moving dots represent joints. .	16
Figure 7 - Velocity along the x,y and z axis of the head during an experiment – vertical lines delineate time intervals where various activities were performed in the following order: Sitting, Getting up, Sitting, Getting up, Lying down, Getting up and Falling.....	18
Figure 8 - Lying down: Distance between the shoulders midpoint and the left knee. The vertical markers identify that start and end time of the “lying down”. Kneeleft d1 to d4 are for individual experiments, the kneeleft avg is the average of all of them.....	19
Figure 9 - Lying Down: Distance between the shoulders midpoint and the left knee now after the normalization and filter. d1 and d2 are for individual experiments, the avg is the average of all of them.	21
Figure 10 - Power transformation - It is observable the peak in power for the 4 trials of the same subject for the sitting activity. These clearly stand out from the background (i.e. before and after the activity change).	22
Figure 12 - Experimental Set: Acquisition Setup	26
Figure 13 - Lying down and Getting up - Relative Distance patterns - demarked areas are related to Lying down and getting up activities respectively	27
Figure 14 - Histogram: Peak Values.....	28
Figure 15 - KinectMotion: System Architecture	33
Figure 16 - KinectMotion: KinectClientAnalysisServices: Class Diagram	34

Figure 17 - Graphical Interface - Client	35
Figure 18 - KinectMotion: Model Flowchart - Future Work	38
Figure 19 - Relative Distance from the shoulders midpoint to selected joints. Demarked areas between vertical lines are related to Lying down and getting up activities respectively	39

List of Tables

Table 1 – State of the art summary	9
Table 2 - Activity Hit Rate – Positive Detections: Sitting, Lying down Falling	29
Table 4 - Sensitivity and Specificity Rate for Sitting 1, Sitting 2, Lying Down and Falling activities and totals	30
Table 3 - Activity Hit Rate – Positive Detections: Getting Up (GU)	30
Table 5 - Sensitivity and Specificity Rate for all the Getting Up (GU) activities and totals.....	31
Table 6 - Precision Rate for all activities together.....	31

1 Introduction

1.1. Motivation and context

The global population is getting older and with this the needs for extra support and monitoring of the aged population grows. At the age of 65 there's an increased chance of 30% to fall during a year time span on for everyday elderly life and this increases to 50% in nursing homes residents where a significant part of these falls require medical attention [1]. Leaving these occurrences unattended might lead to death or lead to high cost medical care and ending any autonomy the elderly had due to irreversible damage.

Recently Microsoft released the Kinect which is a motion sensing input device that features a RGB camera, a depth sensor and a multi-array microphone and this provides for a full-body 3D motion capture. It has a high monitoring potential using its sensors most namely by providing a skeleton segmentation algorithm where it allows for joint position detections on the 3D plane and in real-time.

In our work we propose KinectMotion, a simple monitoring system based on Kinect to support person monitoring and detect occurrence of falls and other relevant events and decrease the probability of being left unattended.

1.2. Objectives

The main objective in this work is to assess the usefulness of the Kinect as part of an easy to deploy and economic system monitoring solution to help the elderly.

Our aim is to use the embedded algorithm and sensors of the Kinect to provide an real-time monitoring system that can detect activities or postures, analyze them and generate a warning so that the subject can be assisted in difficult situations.

1.3. Dissertation structure

This dissertation is divided into the following chapters, excluding this one:

- **Chapter 2 – Posture and activity analysis** presents the bibliographical research done to support this work. Including works on the area of fall detection, home monitoring, posture and activity analysis and video surveillance.

- **Chapter 3 – KinectMotion** presents the model to support the work. It shows, from a design standpoint, the challenges taken so we could achieve the objectives we've defined.

- **Chapter 4 – The changes detection based on the Kinect extracted skeleton** presents the analysis of the raw data extracted from the Kinect and how it was studied, modified and processed to reach the initial objectives of this dissertation. There is also given a step-by-step look at the data to support the choices that were taken in the analysis stage.

- **Chapter 5 – KinectMotion Algorithm Validation** presents the validation process of KinectMotion namely the experimental setup and the results attained

- **Chapter 6 – KinectMotion implementation** presents an overview of the architecture of the system and the technologies used throughout the implementation, providing a full insight of the structure that was developed to support this work.

- **Chapter 7 – Conclusion** presents an analysis to the results, some final conclusions and open issues that may be addressed as future work.

2. Posture and activity analysis

The identification of activities and posture changes is one of relevant task in a monitoring solution, especially those changes that can be associated to relevant events either hazardous (a fall) or that should be followed (e.g. measured activity levels vs. target level). In the monitoring scenario the most common solutions rely on:

- Accelerometers
- Video analysis and quantification

2.1. Accelerometer Approach

A great deal of works in this area focus on accelerometer-based approaches normally in conjunction with other sensors like barometric pressure sensors [3] and other wearable sensor devices. These solutions tend to have great effectiveness on automatic fall detection that most time is coupled with the use of an increasing number of devices is used to track data. This is a problem as placing various tracking devices on a patient can restrict movement leading and increase the deployment time of the overall system besides eventual discomfort of the monitored subject.

Zhang et al., [6] try to minimize the discomfort by using only a belt-worn sensor on the waist. Their algorithm needs a calibration and filtering stage before start using the

acceleration and the orientation sensor to detect falls. Tolkiehn et al., [3] also used a waist-worn belt with the sensors. They use a threshold-based solution. They use a tilt in orientation or a hard collision to detect a fall. They use a tri-axial sensor to determine the orientation of the fall. Gjoreski, Lustrek, and Gams, [2] detect if the subject has fallen based on a setup with 4 wearable sensors throughout the body where the posture recognition is performed using the acceleration of the sensor network. The focus of their work is on determining the optimal accelerometer placement for posture detection. This can be helpful while struggling to discover which points of the human body are important to classify activities. The results are very conclusive and 4 accelerometers can classify nearly 92% of the times the activity correctly.

Bao and Intille, [5] focus on classifying various mundane activities like walking and running and on some more odd activities as brushing teeth or going up an escalator. Working over 5 datasets from 5 different accelerometers they analyzed the G force patterns and were able to classify several activities with an accuracy over 90%.

The main issue of most of these systems is related to daily use, as it is not practical to have an elderly living with the used apparatus and having to set it up every day as placing various tracking devices on a patient might restrict his movement leading to altered data and might also not please him.

2.2. Video Approach

Video based monitoring systems are mostly based on algorithms that either detects the silhouette or the centroid of the subject.

The camera setups used is a relevant detail when discussing the video based posture and activity analysis. Several work use single cameras in posture or fall detection. Simple cameras can be used in posture or fall detection. A single camera can be used as in Foroughi, Aski, and Pourreza, [11] where it detects postures, falls included, by computing an ellipse surrounding the silhouette of the subject to detect the orientation of the person. In this setup instead of simulated environment they used a single camera pointed at a colored wall. Using this information and the position of the head it detects an array of activities with about 91% precision. Boulay et al., [13] focus on producing a blob by

detecting moving pixels in the video as a person. Here they place a camera in a high position in the corner of the room office. Then they split this blob in smaller ones and take the density of these to detect postures like standing or sitting with about 76% accuracy. First off they start applying their technique as a 2D model and later try to apply the model to a 3D environment.

Using the area or the volume of the subject on the footage to track the silhouette is possible to detect sudden changes in orientation and evaluate them. However some of these techniques like the centroid-based approaches may have problem to clearly distinguish a fall, as the estimated velocity changes obtained from the centroid are not accurate enough like in Fu et al. [15]. For this reason these systems often need to be complemented with additional features to be able to classify falls but still often struggle on detecting more subtle changes in postures or activities.

Some systems also use the tracking of the head like Rougier and Meunier, [12] but these need synchronization to initialize the head position so these can't be considered fully automated monitoring systems. The Kinect automatically tracks the head so it doesn't need calibration. It has an automatic synchronization features that bypass this problem.

As for Hu et al., [14] they take the silhouette from a empty room setup with a simple camera placed on the wall and apply a geometrization to the various body parts, converting them to rectangles and afterwards analyze their position, angle and motion. They allow for a parametric study of the human body and motion to extract postures and activities.

There are also systems that rely on multiple cameras. For instance Auvinet et al., [10] using multiple cameras distributed throughout a room near the ceiling it computes the vertical volume of the subject. In this room setup it allows for occlusions as it has various house appliances like a table and sofas. When the majority of the volume of the subject is at a lower height relative to the ground it detects a horizontal orientation of him. A drastic change in orientation points at a fall and it is detected.

2.2.1. Skeleton Approach

The skeleton data is a representation of the human body where selected body part

positions are represented by points in space that are connected according to a geometrical model (some examples presented in Figure 1) Typical skeleton model use joint as reference positions that are connected by rigid body parts (e.g. the elbow is a joint, that connect upper and lower arm). The skeletal representation of the subject can be used to track activities or postures. There are several approaches using skeleton namely the one used by Kinect. The Kinect generated skeleton as seen in the Figure 1, has quite more information that some skeletons used in some of the works shown below.

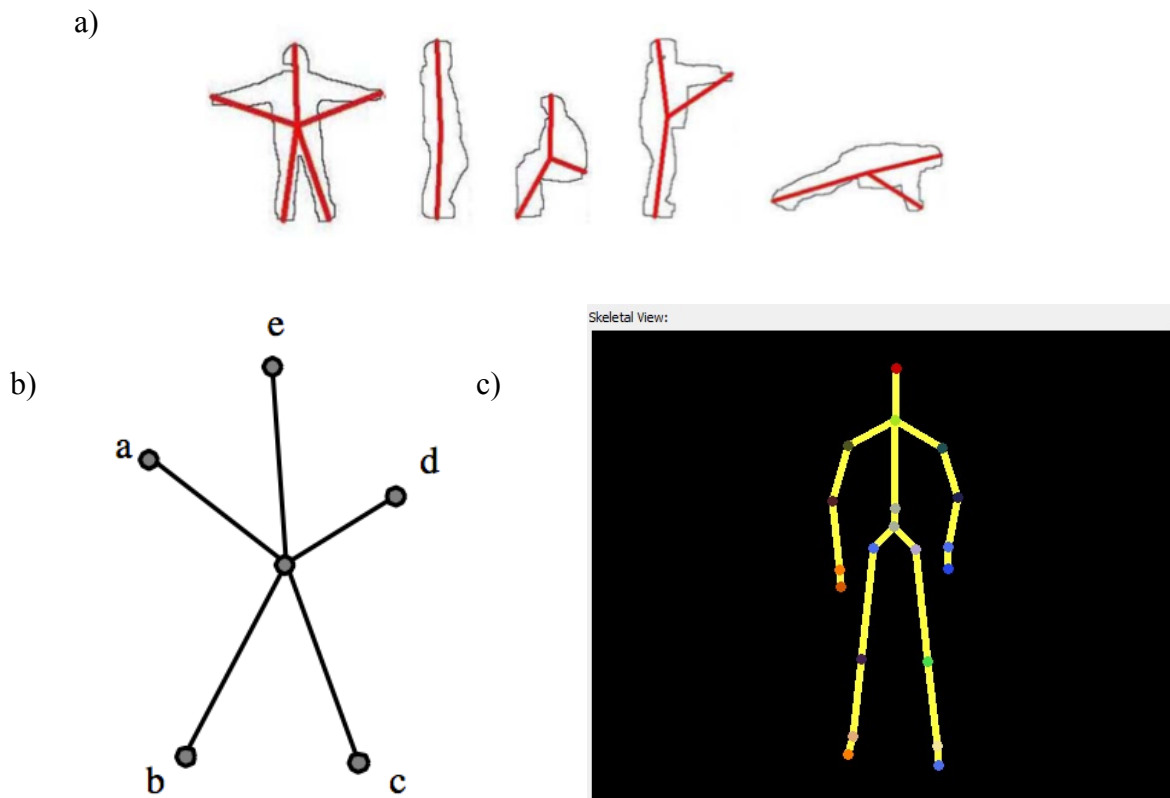


Figure 1 - Examples of Skeleton extracted from video, Kinect included

a) Chun et al. [9] b) Fujiyoshi et al. [8] c) Kinect SDK skeleton

In the Figure 1 there are several examples of skeleton approaches including the Kinect generated skeleton. There are various references to skeleton analysis for human motion as Fujiyoshi and Lipton, [8] where it produces a star skeleton taking the centroid of the subject and then finding the foremost distant point using a distance function from the centroid to the borders. Sometimes it can be hard to judge from the distance as background noise can create edges on the borders and cause it to detect more points than needed. There is a need to smooth out the distances only to detect the peaks necessary for the star skeleton to be created. The key points of the star skeleton besides the centroid are the head,

the upper members and the lower members. Chun, Hong, and Jung, [9] add a 3D component to the star skeleton but the method to retrieve the skeleton is analog with [8].

2.3. KINECT based solutions

Recently Microsoft released the Kinect which is a motion sensing input device that features a RGB camera, a depth sensor and a multi-array microphone and this provides for a full-body 3D motion capture.

The Kinect monitoring features come from a depth sensor, which consists of an infrared laser projector combined with a monochrome CMOS sensor, which captures video data in 3D under any ambient light conditions. Allied with an RGB camera, it supports tracking of up to 6 subjects. It also supports tracking of up to 2 skeletons for gesture-based applications.

There are few works using Kinect to detect activities or postures and we'll focus on trying to use the automated skeleton that is generated like the skeleton approaches we saw before.

2.3.1. Depth Camera Approach

Depth cameras like the one used in Kinect have the immediate advantage that no video recording is taken which ensures the privacy of the monitored people in contrast with video based solutions.

From the few examples found using the depth camera of the Kinect, Rougier et al., [4] aim at detecting falls by locating the centroid relative position to the ground they manage to detect even occluded falls using the acceleration of the centroid during and post-fall. Their system has a nearly 99% accuracy but it lacks the real-time component.

Stone and Skubic, [6] try to use the Kinect Depth Camera in a Web-Cam solution and compare results. In comparison with the old system the Kinect solved some problems on the foreground computing like change of lighting conditions and also reduces the computational requirements that are necessary for robust foreground extraction.

2.4. Summary

In this section we addressed several solutions used to identify posture and activity using different approaches

- Accelerometers
- Video
- Kinect

Some brief conclusions are possible at this point that inspired some of the decisions taken along this project.

- Accelerometer-based systems require a setup that might be hard for the elder to setup alone and might be uncomfortable. Also considering the mental faculties of the subject the setup can sometimes be easily be forgotten and completely disregarded. This may hinder the practical adoption of this solutions
- While video provides several solutions using the video as main data input places some privacy issues and, although effective, some of the presented solution do not provide reliable online solutions
- Kinect based solutions are low-cost and address the privacy issues and online monitoring by depending on consumer level implementation provided by Microsoft namely the depth camera and automated skeleton extraction that “hide” the details under a reliable and tested API/framework. However it is still lacks concrete examples that support its effectiveness.

In table 1 we present a resume of all works addressed.

Ref	Accuracy	Activities Recognized	Subjects	Type
Bao et al. [5]	>80%	Walk, Run, Sit, Standing Still, Other	20	A
Tolkiehn et al. [3]	≈86%	Directional Fall, Sit, Walk, Lie Down, Others	12	A(1)
Zhang et al. [6]	100%	Walk, Fall, Standing, Sit	10	A
Gjoreski et al. [2]	[48%,92%](2)	Trip, Fall, Sit	11	A
Rougier et al. [12]	-	Fall	1	V
Auvinet et al. [10]	[80%,100%](3)	Fall	-	V
Foroughi et al. [11]	≈91%(7)	Walk, Run, Stumble, Various Falls	50	V
Boulay et al. [13]	76%	Standing Postures, Sitting Postures, Bending	-	V
Hu et al. [14]	-	Silhouette Tracking	-	V
Rougier et al. [4]	≈99%	Falls (occluded and not occluded)	-	K
Fujiyoshi et al. [8]	[90%,98%](4)	Motion Classifying	-	V
Chun et al. [9]	-	Postures, Standing, Sit	-	V
Stone et al. [7]	-	Comparison: Normal Cameras with the Kinect(5)	-	K
Fu et al. [15]	-	Fall, Stand, Walk, Crouching	-	(6)

Table 1 – State of the art summary

A- Accelerometer; V- Video; K- Kinect

(1)Barometric Pressure Sensor

(2)Different Experiments

(3)Depending on the number of cameras

(4)Depending on the threshold frequency

(5)They accessed the Kinect Depth Camera capabilities against normal web cams.

(6)Visual Sensor. Method comparison.

(7)Uses precision to rate

3. KinectMotion

The main objective in this work is to assess the usefulness of the Kinect as part of an easy to deploy and economic system monitoring solution to help the elderly. Our aim is to use the embedded algorithm and sensors of the Kinect to provide a real-time monitoring system that can detect activities or postures, analyze them and generate a warning so that the subject can be assisted in difficult situations.

In this section we describe our solution – KinectMotion. In KinectMotion we used the Kinect Skeleton raw data stream obtained through the Kinect SDK to perform online detection of activity and posture changes and provide remote access to the same information namely to store the data for further analysis or generate warnings.

We assumed some restrictions namely the scenario: a room size with a bed, chairs and table – typical living space of a typical room similar to used in other works as [10]. The point is to reproduce a normal living space with normal elements and use them in the experiment.

In this section we'll present the system architecture and the system requirements.

The main requirements that conditioned the design:

- Remote monitoring capabilities
- Activity recognition
- Data persistence for posterior data analysis

3.1. The overall concept

KinectMotion conceptual model is presented in Figure 2. The basic idea is that from the data acquired from Kinect that will analyze and process it in order to infer/detect different postures/activities using some pattern matching solution. From the results warnings and/or activity classification can be produced. The online monitoring is achieved by repeating this same loop over and over.

In KinectMotion our focus will on processing the skeleton stream delivered by Kinect that together with the geometric model will be used in the activity and posture analysis.

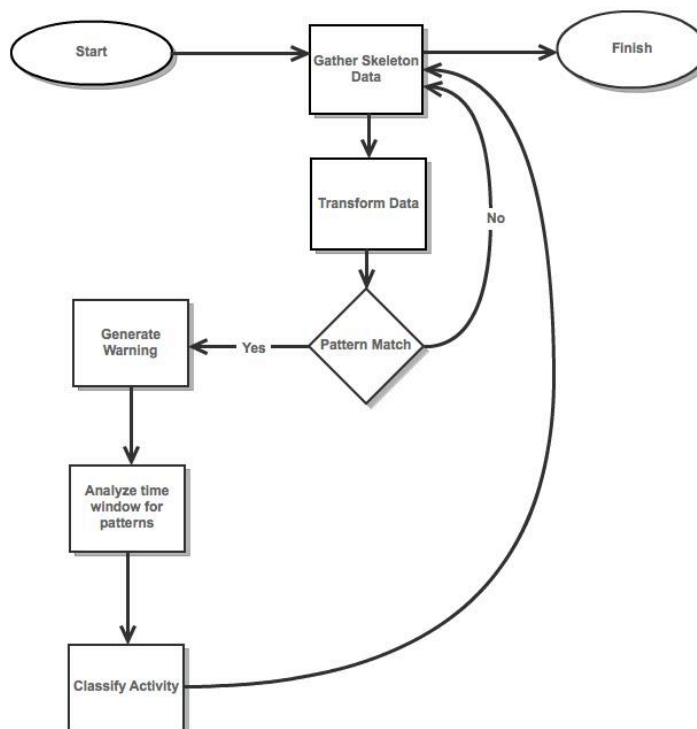


Figure 2 - Conceptual flowchart

This flowchart represents the initial concept for this system. As the developing process started we decided to focus first on the activity detection, the first 4 blocks of the chart.

3.2. Gather skeleton data

The skeleton information obtained from Kinect was our main and only data source used in KinectMotion. This information consists on information on the coordinate of each joint in the Kinect skeleton model (Figure 3.).



Figure 3 - Skeleton Frame

The Kinect can detect up to 20 joints and by linking them together it produces a skeleton and provides upgraded star skeleton featuring various midpoints in the upper and lower members and various points on the torso.

For each joint, Kinect is able to provide the information of the respective coordinates (X,Y,Z) along time. In Figure 4 we present a sample of a “text” version of the Kinect’s skeleton stream used.

```
Time: 05-03-2012 15:27:48.287 - JointID: HipCenter - Position: W: 1 , X: 0,674933 , Y: -0,2372757 , Z: 2,189822
```

```
Time: 05-03-2012 15:27:48.298 - JointID: Spine - Position: W: 1 , X: 0,6749511 , Y: -0,1707318 , Z: 2,167748
```

```
(...)
```

Figure 4 - Raw Data – A fragment of a the data available in a skeleton frame with joints associated coordinates and acquisition timestamp

The time was also set for the actual time the experiment took place and so for each data set we've converted the time of the first frame and set it as the initial time for each experiment.

3.3. The Kinect Skeleton referential

The raw data provided for each joint was in a referential (Figure 5.a)) but the X-axis and the Y-axis provided values between -1 and 1 and the (0,0) coordinate was the center of the screen. To work in with pixels we considered a 640 for 320 pixels screen and transformed this data into a referential by placing the (0,0) coordinate in the bottom left screen and set the range from the coordinates from X:(-1,1) to (0, 640) and Y:(-1,1) to (0, 320) as shown in Figure 5. b).

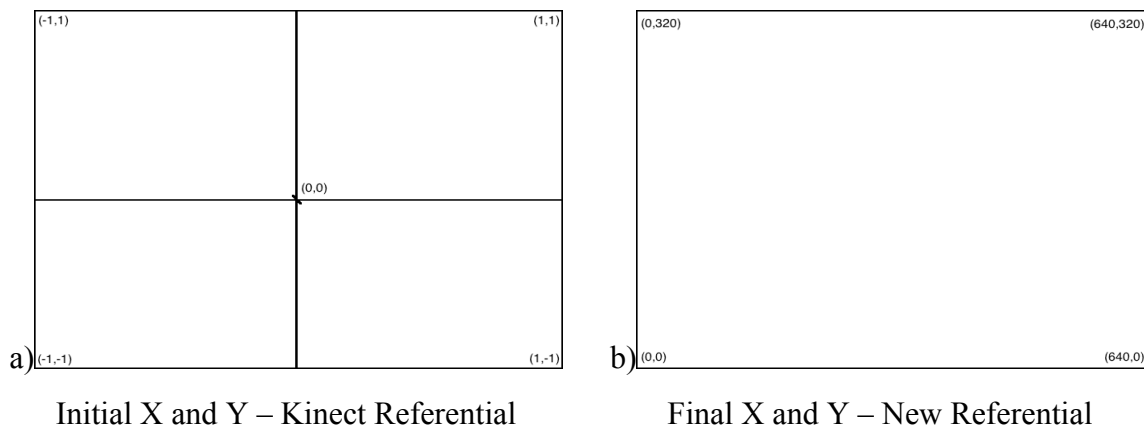


Figure 5 - Referential transformation – X and Y coordinates transformation

We considered a frame to be a set of joint related data with the same timestamp joined together. This was used to establish the frame-rate.

4. The changes detection based on Kinect extracted skeleton

In KinectMotion, our aim is to detect posture and activity changes. We considered the following activities/postures:

- Falling
- Sitting
- Lying down
- Getting up

Although a final method will be presented in the end of this section we opted to describe the process lead to it, while describing the several steps (some unsuccessful) and decisions taken along the process.

In the falling activity we've asked our subjects to take a sideways fall into a mattress, stay on the floor a couple of seconds and then get up. The Kinect captures this event at a slightly oblique angle.

The lying down activity is quite similar to the falling activity with the difference that the subject lies down to his back on the sofa next to the mattress. The angle of the capture is also slightly oblique. The subject is also required to stay lying for a couple of seconds and then get up.

In the sitting activity we've set up 2 scenarios. The activity is the same but the angle of

capture is different. On the first time we capture the activity with the subject facing directly the Kinect and on the other on we capture the activity at a 90° angle.

4.1. Transform Data

Our first concern was to extract relevant information from the Kinect generated skeleton with 20 joints spatial positions were used as pixels coordinates.

4.1.1. Noise

During a preliminary analysis, support also on visual inspection of both video and skeletons we observed that using the Kinect skeleton we had a much noise in several situation namely:

- Some of the positions the subjects took occluded some members
- The subject were not in positions according to SDK model was not design to detect which led to get high velocity of some joints even when the subject was standing still

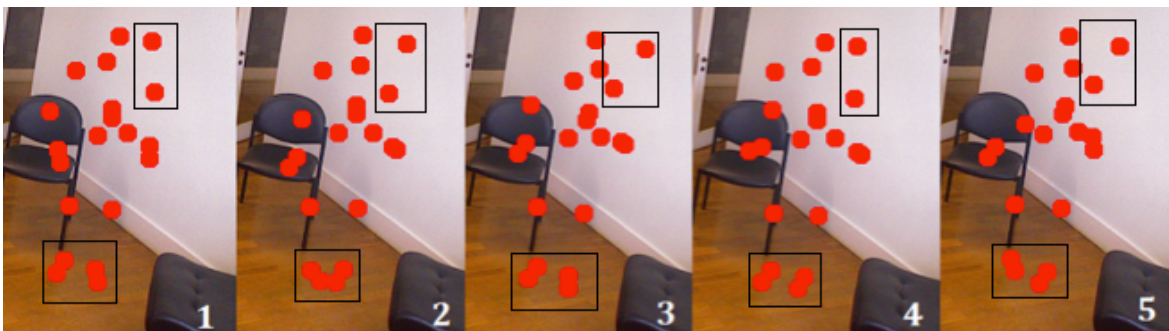


Figure 6 - In Kinect skeleton data we observed that non-moving parts presented changes. In the example the moving dots in the highlighted area in consecutive frames show changes although the subject position did not change. The moving dots represent joints.

In Figure 6 even though the subject was completely still and these frames are sequential the highlighted areas show that the joints have an irregular behavior, which is from where most of the noise comes.

4.1.2. From pixels to meter

We performed a simple conversion from pixels to meters to abstract from the “pixel” measures. Using the average size of the subjects of the experiment and the average size of the torso of a person of that height, we estimated that 29.3 pixels were about equal to a meter based on the torso size of the average subject. The measure estimation was performed at the starting position of the experiment. This method, although simplistic allowed for a generic way to convert all measures into the same referential that accommodated the differences between all the subjects.

4.1.3. Velocity extraction

Based on the 20 Hz skeleton frame rate (0.05 on the formula) and difference between the Joint position (S_{xt}) and the position of the anterior joint ($S_{x(t-1)}$) in pixels, we estimated the velocity for each of the joints, in meters per second, using the following formulas;

Velocity in the X-axis:

$$V_x(Joint)_t = \left| \frac{\left(\frac{S_{xt} - S_{x(t-1)}}{0.05} \right)}{C} \right|$$

Pixels to Meter constant:

$$C = \text{pixelPerMeter} = 29.3$$

Vectorial Velocity:

$$V_v(Joint)_t = \sqrt{V_{x_t}^2 + V_{y_t}^2}$$

Taking the Velocity in the X-axis and in the Y-axis we calculated the Vectorial Velocity also in meters per second.

In Figure 7 we present a graph presenting the velocity (absolute value, X-axis and Y-axis) of the heads along one single experiment. From a quick observation it is reasonable to assume that for each of the activities considered (Sitting, Getting up, Sitting, Getting up, Lying down, Getting up and Falling) may have different velocity signatures in accordance with our intuitive impression.

In our preliminary analysis on the velocity patterns, we found that velocity information was noisy especially in time intervals with jitter where Kinect skeleton presented variations that were not easily distinguishable from relevant activities.

Note that in Figure 7 during some static postures the velocity was not zero supporting our initial observations on the presence of noise in the data.

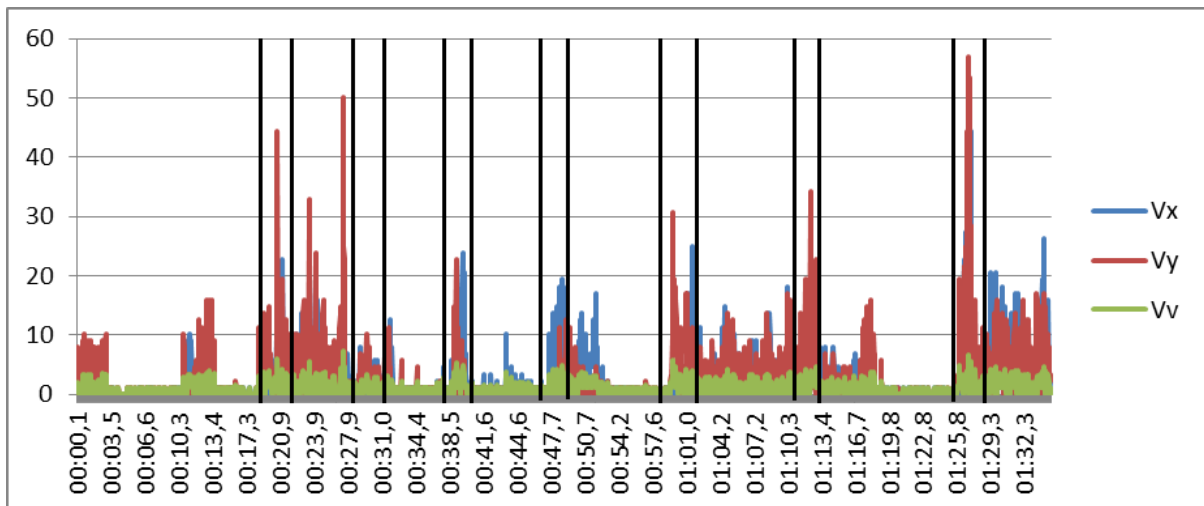


Figure 7 - Velocity along the x,y axis of the head during an experiment – vertical lines delineate time intervals where various activities were performed in the following order: Sitting, Getting up, Sitting, Getting up, Lying down, Getting up and Falling

4.1.4. Relative Distance between joints

Besides velocity we explored the relative positions of the body parts (distances) in order to describe / identify patterns that could be mapped into activities like sitting, falling or lying down and getting up after each on of these.

As starting point we selected the midpoint between the shoulders that besides being in the torso it provided a good fixed related reference in relation to the moving body parts (e.g. arms, legs, head). From empirical observations it also seemed to be a reliable reference to use even when in present of noise in the Kinect extraction

With the relative distance we found was suggestive patterns for activities like seating, falling or lying despite the presence of noise. This was observed in both the analysis of the individual and average distance profiles as exemplified in Figure 8. In our analysis we also considered other distances between the shoulder midpoint and other joints like knee and hip.

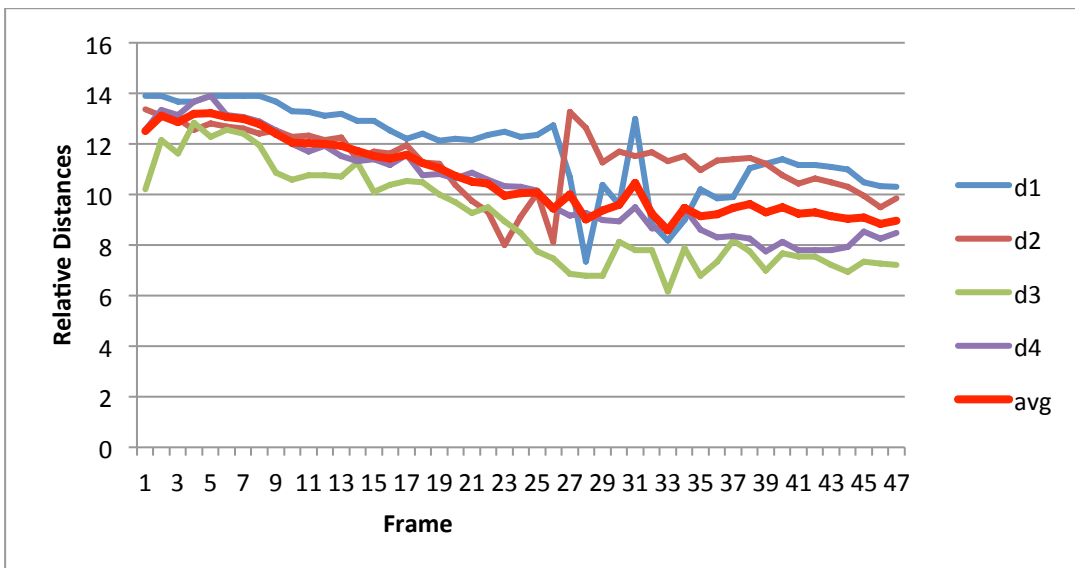


Figure 8 - Lying down: Distance between the shoulders midpoint and the left knee. d1 to d4 are for individual experiments, the avg is the average of all of them.

On the Figure 8 it's quite visible that there's a pattern in the transition where we there is a sudden drop of the given distance. This is clear even taking in account the jitter of Kinect

and the different postures of the subject after lying down.

To extract the relative distance patterns, in our algorithm we performed some preprocessing.

To extract the distance between Joint (joint) and Intermediate point of the shoulder's (Inter) joints we used the standard distance (in pixels):

$$dInterJoint = \sqrt{(joint_x - Inter_x)^2 + (joint_y - Inter_y)^2}$$

We normalized the distance values (dInterJoint) using as reference the prior 1 second time window (20 frames) to estimated both the mean (meanJoint) and the standard deviation (stdJoint). The normalized distance (normalDJoint) was calculated using the following expression:

$$normalDJoint = \frac{dInterJoint - meanJoint}{stdJoint}$$

As a result we obtained normalized distances, in an arbitrary unit, that were comparable among different subjects, after normalizations.

A simple low pass filter was applied to reduce the “noise” – in this case the sharp and high frequency changes. The filter parameters were selected based on an iterative process based on the observation and analysis of several events acquired with one of the subjects. Our aim was to obtain a smooth power peak related with the map that was distinguishable from the background (e.g. absence of relevant event).

$$\begin{aligned} filteredDJoint_{t=0} &= normalDJoint_0 \\ filteredDJoint_{t>0} &= normalDJoint_t * alpha + normalDJoint_{t-1} * (1 - alpha) \\ alpha &= 0.1 \end{aligned}$$

For each data sample (normalDJoint) we obtained a filtered sample (filteredDJoint), also in an arbitrary unit. The alpha value considered was 0.1. The Figure 9 shows the some

of the data after the transformations for the same activity as in Figure 8.

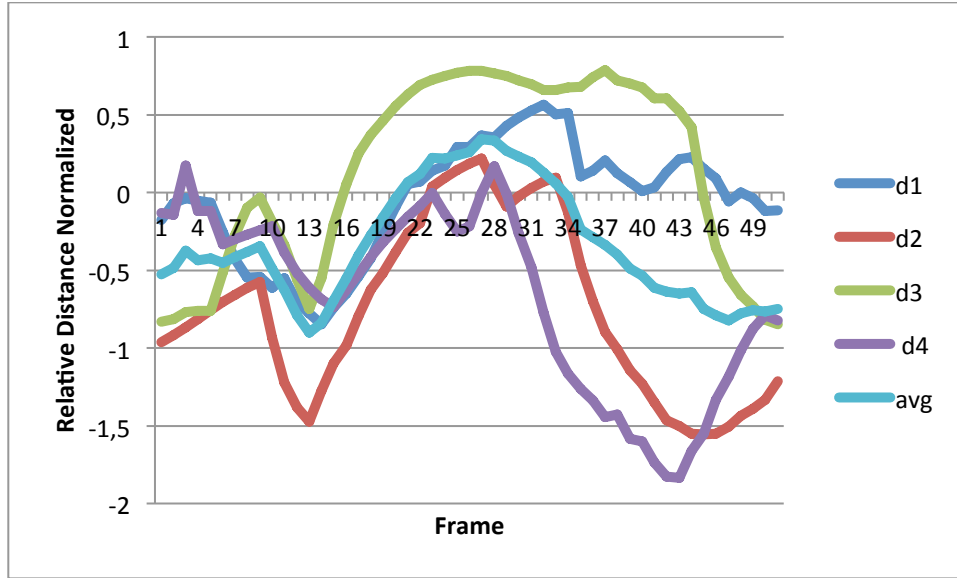


Figure 9 - Lying Down: Distance between the shoulders midpoint and the left knee now after the normalization and filter. d1 to d4 are for individual experiments, the avg is the average of all of them.

4.2. Pattern Match through thresholds

Based on the observed “slope” in all of the activities we used a common approach to detect transients in signal processing – a power transform (square of the values). Power transform enhances and amplifies changes found in the data regardless of the signal /trend (i.e. increase or decrease).

We used the sum of the squared normalized distances where *filteredDJoint* represent the sample for each of joint:

$$Sum = \sum filteredDJoint^2$$

Sum is represented also in an arbitrary unit.

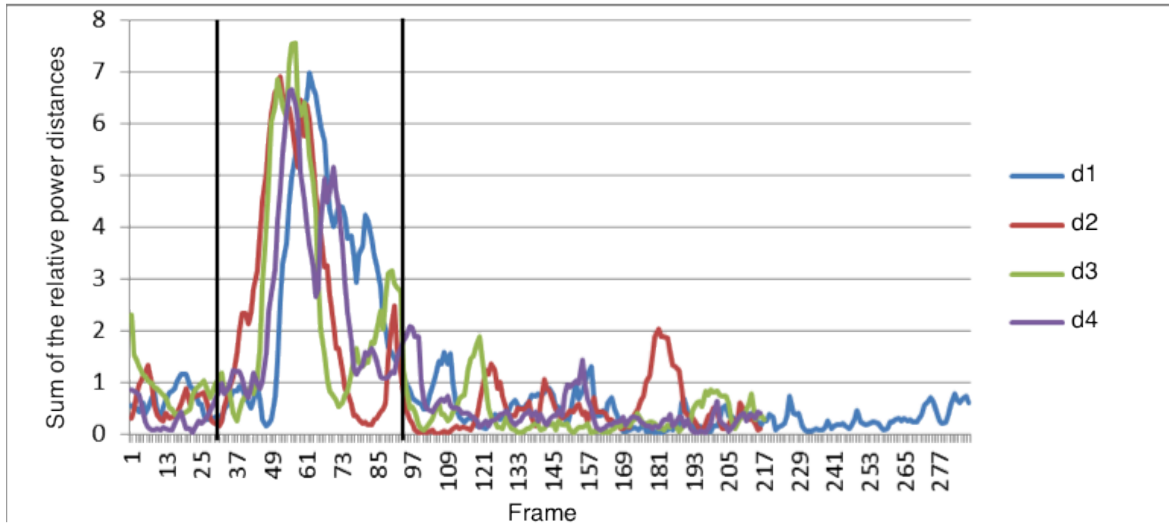


Figure 10 - Power transformation - It is observable the peak in power for the 4 trials of the same subject for the sitting activity. These clearly stand out from the background (i.e. before and after the activity change).

In Figure 10 we present a good example of the benefit of using the power transform. The peak of the power is clear and stands well above the background power values. At the same time it is intuitive to relate it with the activity event marked by the vertical lines in the figure. This pattern was also observed in all other analyzed activities and as so we can apply thresholds to the values and detect activities as it was pretended.

This observation led us to the hypothesis that by using simple threshold over the power values we could be able to find the occurrence of relevant changes of activity and postures.

4.3. The final model

KinectMotion posture and activity change detection algorithm is based on establishing threshold over a power based representation of the sum of the distances between the skeleton reference point – the shoulder midpoint and the following reference joints selected previously. The relevant changes exhibit clear power peaks and, in the algorithm we selected all power peaks above the selected threshold.

Although the process to achieve this solution was a result of several iterations, the final

algorithm is simple and can be easily tuned. By establishing threshold over the power values we can tune the sensitivity and specificity of the algorithm to detecting changes in postures and activities namely, sitting, lying down, falling and getting up.

Here follows a summary of the full steps (Figure 11):

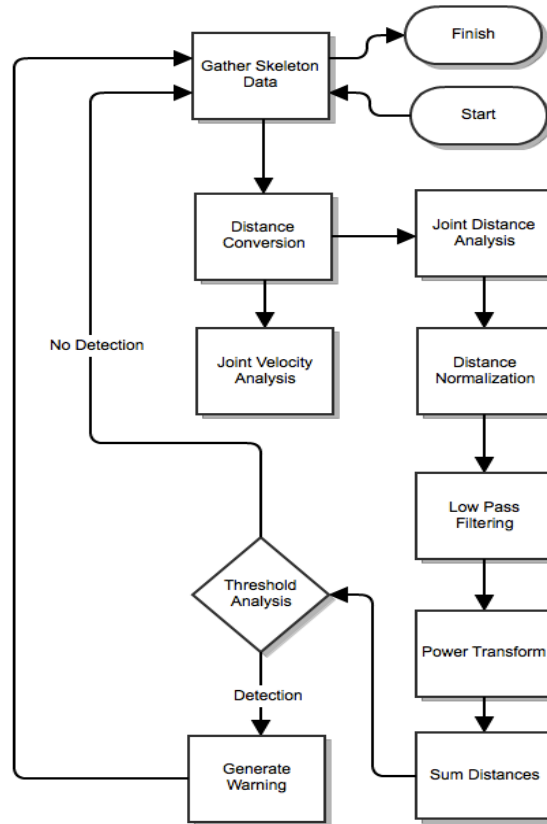


Figure 11 - KinectMotion: Final Model Flowchart

5. KinectMotion algorithm validation

In this chapter we describe our assessment and validation of the proposed activity change detection algorithm that we use in KinectMotion.

5.1. Data and methods

For supporting our work and validation we performed an experimental procedure that was performed by a total of 6 subjects that were monitored using the Kinect.

In the acquisitions we acquired the Kinect skeleton that was used as our main analysis input. All this data was stored and used in our search for the activity and posture algorithm for KinectMotion.

5.2. Subjects

There were 6 volunteers that went through the experiment with 5 repetitions. Their height ranged from 1,65 to 1,80 centimeters and weighted from 60 to 80 kilograms, the average age was 23.

5.3. The procedure and the experimental protocol

The subjects were asked to perform the following activity sequence in the setup depicted in Figure 6: 1-P1, 2-M1, 3-P1, 4-P2, 5-M1, 6-P1, 7-P2, 8-M1, 9-P1, 10-P3, 11-M1, 12-P1, 13-M2, 14-P3, 15-P1 (Figure 12). This sequence takes various activities extended throughout the entire experimentation room. The Kinect was held at an approximate height of 2 meters with a slight incline. The placement of the Kinect when using the system later is important as it should have an ample view over the room it is supposed to monitor and never more than 4 meter away from the zone it should watch over since the range of detection is roughly between 1 and 4 meters.

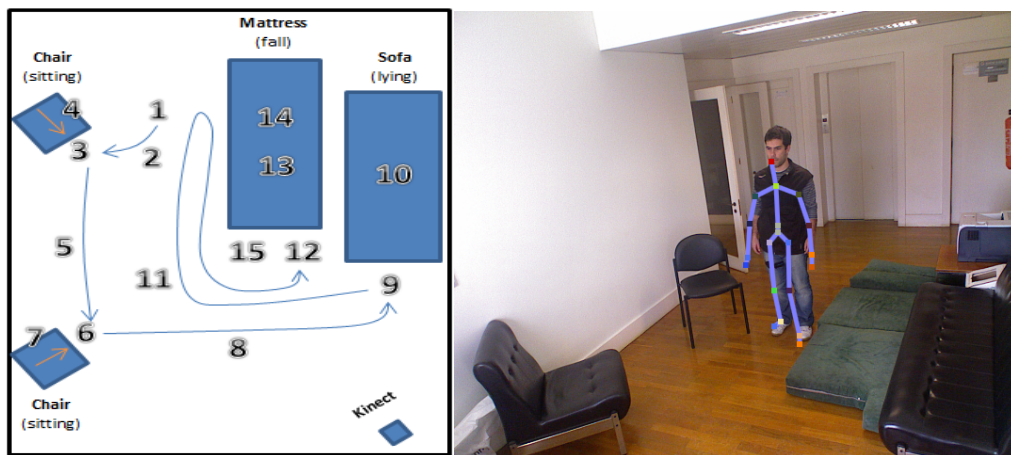


Figure 12 - Experimental Set: Acquisition Setup and acquired video frame with skeleton overlay using Kinect

Our focus is on the following transitions from 3 to 4 as Sitting 1, 6 to 7 as Sitting 2, 9 to 10 as Lying Down and the activity 13 as Falling.

5.3.1 Getting up

Although not initially planned we also analyzed the “getting up” activities that occur after the main activities as we observed, that besides the higher heterogeneity (e.g. getting up after lying down vs. getting up after sitting), they also might represent worthy events to

monitor namely to detect the recovery or not of the monitored person. From the relative distance patterns we found patterns prone to be detected by our method. A good example is given in Figure 13.

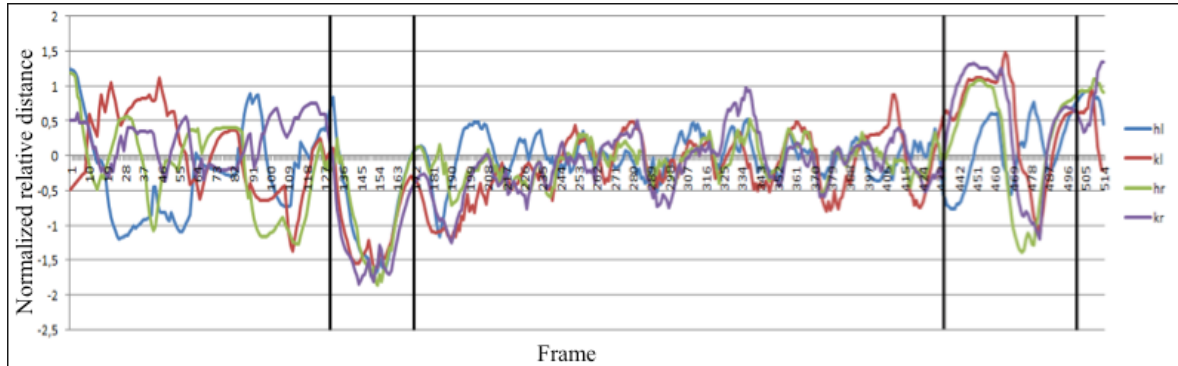


Figure 13 - Lying down and Getting up - Relative Distance patterns - demarked areas are related to Lying down and getting up activities respectively. hl, kl, hr and kr stand for the distance between reference point on the chest to the left hip, left knee, right hip and right knee respectively.

5.4. Model evaluation

From our preliminary analysis it was clear that it would be difficult to achieve a universal parameterization suitable for all subjects and population. So our main aim was to understand the impact of the setting different thresholds in both sensitivity and specificity of KinectMotion algorithm.

We considered 5 power threshold values based on preliminary assessment over the data. The data sets created after the power transform showed that the peaks related to activities had values between 5 and 14. By analysis of the Figure 14 the majority of the values are located between 5 and 9, so we choose 5 thresholds with values from 5 through 9.

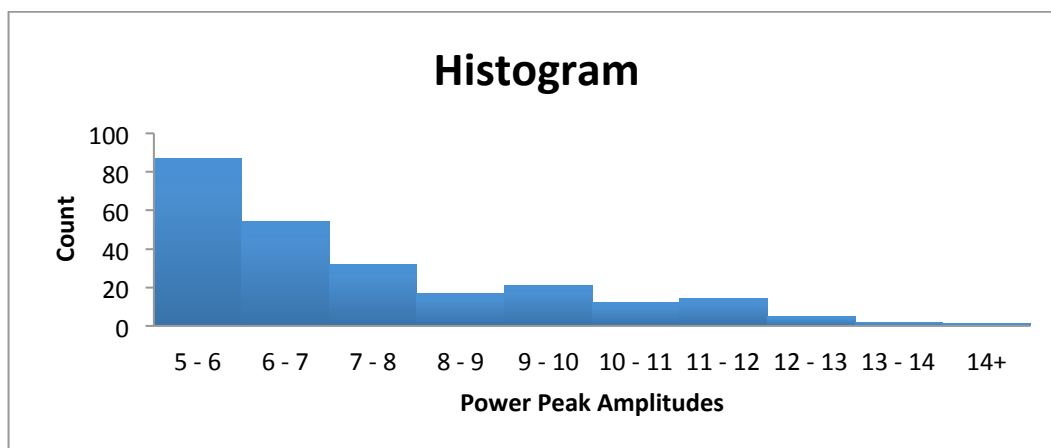


Figure 14 - Histogram: Peak Values

In this evaluation we took 18 experiments from the various subjects to a total of 53382 frames analyzed and had 144 activities to detect: 36 sitting down activities (18 facing the Kinect and 18 with an 90° angle to the left), 18 lying down activities, 18 falling down activities and 72 getting up activities (getting up after the other activities).

Considering that we are detecting a time point and our aim is to detected an interval in time where a given change occurred we considered a hit whenever a peak fall into a given activity interval.

5.4.1. Sensitivity and specificity

To describe the results we're going to use Sensitivity, Specificity, Precision and Accuracy that gives out a better grasp at the quality of the model. We'll use the following notation:

- True Positive – Activity correctly detected
- True Negative – Lack of Activity correctly detected
- False Positive – Activity incorrectly detected
- False Negative – Activity not detected

The sensitivity tests the model's ability to identify positive results.

As opposed to Sensitivity, Specificity tries to test the model’s ability to identify negative results, which means the ability to not wrongfully detect an activity.

Precision measures the proportion of the positive population rating the quantity of real hits out of all the population of hits.

$$Sensitivity = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$Specificity = \frac{True\ Negatives}{True\ Negatives + False\ Positives}$$

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

On the Specificity and Precision we only look into the total values, as we can’t associate the False Positives to a certain activity.

Were we are going to show the sensitivity of each activity and the global sensitivity.

5.4.2. Results

We have split the results in two parts: one is addressing the main activities and the second addressing the “getting up” action. The results of the tests combining different threshold and different activities are presented below.

Here follow the results from the activities:

		Activity	Sitting 1	Sitting 2	Lying down	Falling	Outside	Totals
		Total	18	18	18	18	53238	72
Thresholds	9	Hits	10	0	13	15		38
	8		12	2	16	15	1	45
	7		15	8	18	17	3	58
	6		18	11	18	17	7	64
	5		18	16	18	18	28	70

Table 2 - Activity Hit Rate – Positive Detections: Sitting, Lying down Falling

		Sitting 1	Sitting 2	Lying down	Falling	Totals	
		Sensitivity					Specificity
Thresholds	9	55,56%	0,00%	72,22%	83,33%	52,78%	100,00%
	8	66,67%	11,11%	88,89%	83,33%	62,50%	100,00%
	7	83,33%	44,44%	100,00%	94,44%	80,56%	99,99%
	6	100,00%	61,11%	100,00%	94,44%	88,89%	99,99%
	5	100,00%	88,89%	100,00%	100,00%	97,22%	99,95%

Table 3 - Sensitivity and Specificity Rate for Sitting 1, Sitting 2, Lying Down and Falling activities and totals

Considering the sensitivity values we can see that the activities Sitting 1, Lying down and Falling present good rates, meaning that the system will almost always detect these activities if we place the threshold in a value between 5 and 7. The rates lower on the activity Sitting 2 and can be related to the angle of the capture or can be related to the chair, which was different from the activity Sitting 1. Even so only the higher valued thresholds present lower sensitivity values and thresholds between 5 and 6 provides sensitivities above 60% and don't negatively affect the global rates. Globally from the analysis of the sensitivity values it is reasonable to conclude that this algorithm may provide a realistic options to detect relevant activity changes with low number of false detections as the observed specificities are never lower than 99%.

The high specificity values show that the population of misjudged hits is really low when there is no activity happening.

Here follow the results from the “getting up” activities:

		Activity	GU1	GU2	GU3	GU4	Totals
		Total	18	18	18	18	72
Thresholds	9	Hits	0	0	0	0	0
	8		1	1	0	0	2
	7		3	1	1	1	6
	6		9	4	5	4	22
	5		14	1	13	8	45

Table 4 - Activity Hit Rate – Positive Detections: Getting Up (GU)

		GU1	GU2	GU3	GU4	Totals	
		Sensitivity					Specificity
Thresholds	9	0,00%	0,00%	0,00%	0,00%	0,00%	100,00%
	8	5,56%	5,56%	0,00%	0,00%	2,78%	100,00%
	7	16,67%	5,56%	5,56%	5,56%	8,33%	99,99%
	6	50,00%	22,22%	27,78%	22,22%	30,56%	99,99%
	5	77,78%	55,56%	72,22%	44,44%	62,50%	99,95%

Table 5 - Sensitivity and Specificity Rate for all the Getting Up (GU) activities and totals

Although our initial impressions were that the “getting up” activities would also present a clear power signature as we found in the activities, this did not corresponded with the obtained results. As the tables 2 through 6 reveal, the method used in normal activities is not adequate for these. Although presenting high specificity when detecting a “getting up” the sensitivity is quite low when using threshold for the normal activities values considered (below values of 77%). From a visual inspection of some of the results it was clear that "getting up" activities still presented a peak in the power transform representation but these were lower than for normal activities and less distinguishable from the background. The higher sensitivities with lower threshold support this interpretation.

On assessing the precision of our algorithm we considered activities and getting up actions altogether, as both are relevant events in the KinectMotion regardless of their nature. Otherwise, by excluding them, they would be considered misjudged hits that would bias the results, even when though the sensitivity values for the Getting Up activities are low.

		Total Precision
Thresholds	9	100,00%
	8	97,92%
	7	95,52%
	6	92,47%
	5	80,42%

Table 6 - Precision Rate for all activities together

The precision values shown in the Table 6 show that the overall precision of the system is high(over 80% for all considered thresholds) with this method out of all the detections most of them were correctly evaluated. The results for threshold 5 suggest that it may already be more prone to be affected by lower amplitude power peaks. Further analysis needs to be done to determine if these are relevant information or just noise in relation with the purpose of this algorithm - detect normal activities.

5.4.3. Conclusions

KinectMotion algorithm presents good specificity and sensitivity trade-offs when using same threshold along the tested population. As the data used is normalized we can use the same threshold between all subjects. Our results suggest a threshold should be between 6 and 7 as it is a good compromise for high specificity and high sensitivity.

Naturally these results are only indicative and limited to a group of 6 healthy subjects within a very specific setup. Nevertheless, they seem to suggest that such a simple model may be useful in real scenarios with the adequate customization namely on the setup, on the population.

6. KinectMotion implementation

In this section we'll explain the system implementation details as the architecture, components and also how that communicate with each other and work. There are two essential blocks on KinectMotion, which has got a basic client-server architecture. The server side is where the Kinect is connected and does the skeleton tracking. It consists of a PC and sends the information to another PC through TCP/IP. That other PC is where the client is situated and it receives the skeleton tracking data that the server sent and processes it as explained on the section before. On this PC is also where the persistence of the data is kept and it is there that the warnings are generated. The Figure 15 shows a rough representation of what we mentioned.

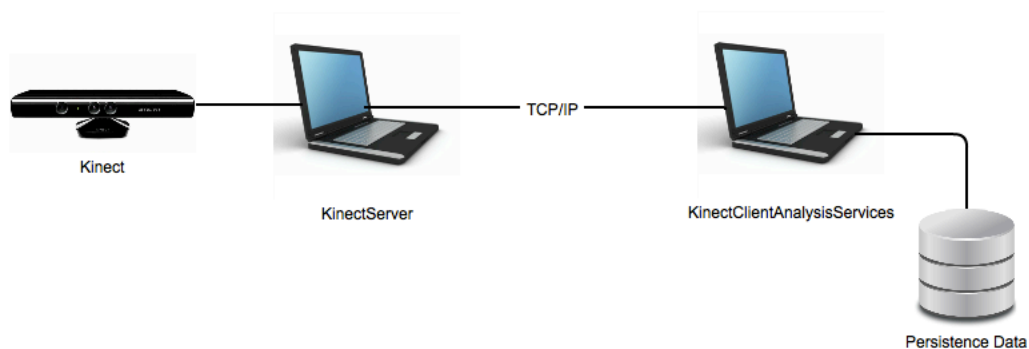


Figure 15 - KinectMotion: System Architecture

6.1. KinectServer

In this component we used the Kinect SDK runtime to access the skeleton frame stream that is generated when it detects a subject. It generates an event and then we access the data. Then we open a socket connection that waits for a connection to start flushing the data from the skeleton frame stream, allowing for the analysis to be made either on the same machine or through the web. This server also has an interface, which shows the skeleton in real-time. By accessing the skeleton detected by the runtime simply by sending a socket with that information is enough. The point of the server is only to generate a skeleton and send it. This component is made out of C# as the SDK only allows it.

6.2. KinectClientAnalysisServices

This component takes all the raw data from the files. It has a replay interface so that the experiments can be viewed and has the various data structures to calculate and generate the warnings and propagate them. It also generates the support files, which can be used as backup for the data. These files are stored can be accessed and viewed on the replay interface for further analysis.

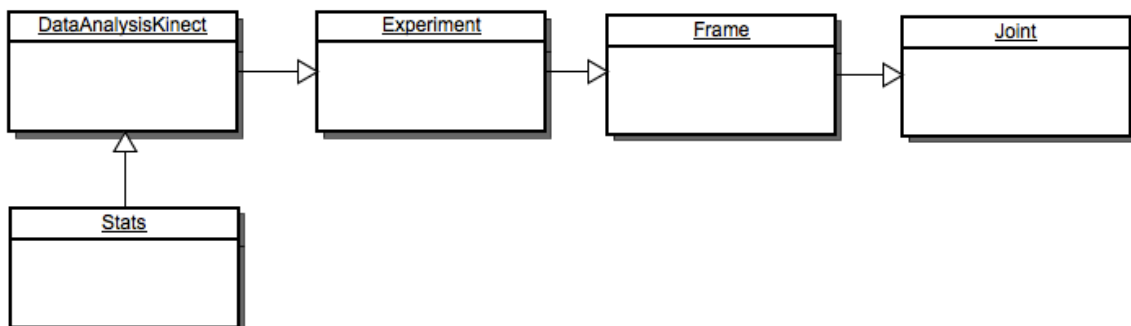


Figure 16 - KinectMotion: KinectClientAnalysisServices: Class Diagram

The data received from the KinectServer is parsed and placed in a data structure called Experiment. This data then is divided into various Frames, which is a representation of all the points of the skeleton at a given time. These points are called Joints, which contains the useful information as the time the point was captured, his name and his coordinates and the class diagram related to this implementation is represented in Figure 16. The notion of Frame is important in the viewer as the Analysis Client launches a controllable thread which shows the Frames at a given time rate in resemblance to a stop motion. If the target of the analysis client is a stored file the interface also works as a replay system with play, stop, fast-forward, slow-motion and a go-to option. The Figure 17 shows the aspect of the graphical interface. The Stats class provides with various mathematical functions as the calculation of Means and Standard Deviation for the data treatment.

The KinectClientAnalysisServices is made in Java in an effort for a multiplatform system, as only the Server is made out of C#.

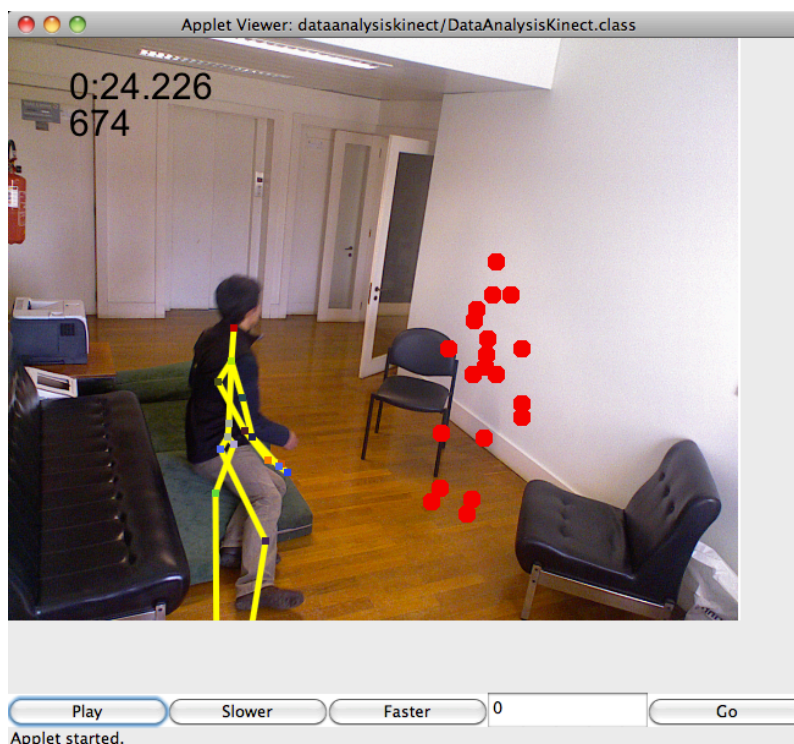


Figure 17 - Graphical Interface – Client: The numbers represent the running time window and the actual frame and the red dots represent the skeleton being tracked.

6.3. Communication

The Kinect Server and the KinectClientAnalysisServices communicate through TCP/IP systematically sending status strings through sockets. On the server side it opens a socket on the port 4000 that the client must access. When there's a connection it joins the skeleton runtime and while the server accesses the skeleton stream to draw the joints on the graphical interface it also writes the data to be sent through the socket, each time a new joint is analysed.

Data String:

```
"Time: " + sec.ToString() + "." + milisec + " - JointID: " + joint.ID + " - Position: W: " + joint.Position.W + " , X: " + joint.Position.X + " , Y: " + joint.Position.Y + " , Z: " + joint.Position.Z + "\n"
```

To access the server you only need to open a socket on the server IP, on the port 4000 and receive the data String shown above.

7. Conclusions

The focus of this dissertation was to assess the role of Kinect as inexpensive monitoring tool with capabilities to detect changes in activities and postures. From our results we can conclude that it is a realistic solution to detect changes in postures, especially situations that need to generate warnings.

The high precision values obtained by KinectMotion algorithm suggest that using Kinect we have the potential to provide great tool to detect activities and it can be used in a home monitoring environment to detect relevant changes like falls. In our system, falls detection showed high sensitivity values. The Kinect has the potential to be used as a robust monitoring appliance and be used at home, hospitals and nursing homes.

7.1. Future Work

Naturally, considering that the scope of this work is support elderly monitoring further study must be done as our studied population was made from healthy and young subjects. It's almost a certainty the elders will present different patterns although we still believe that these will also have signatures prone to be classified in both power and skeleton based distance representations.

Our work detected relevant changes in posture and activities, clearly in the future we should try to classify the activities related with the change we detected. As the Figure 18 represents:

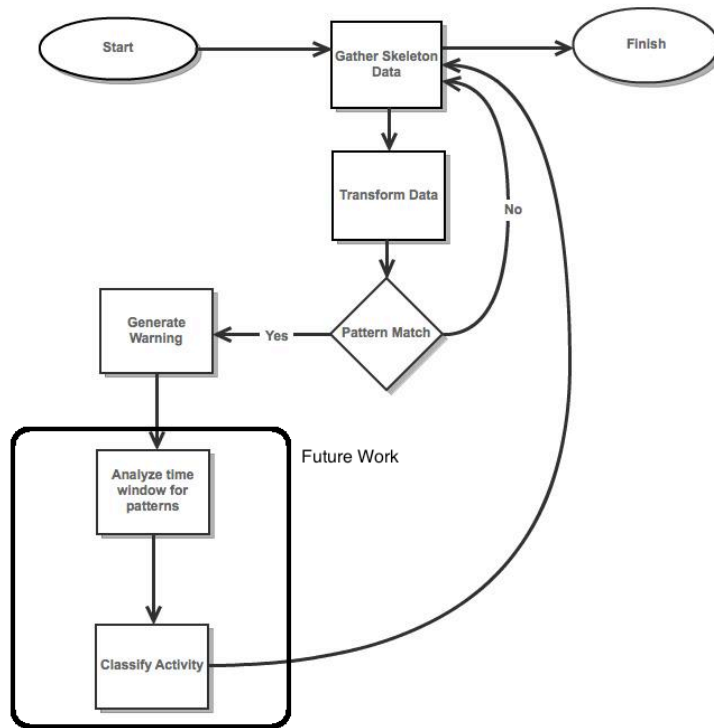


Figure 18 - KinectMotion: Model Flowchart - Future Work

From our analysis, it was also clear after visual and empirical observation that the relative distance information seems to have enough information to perform an activity classification.

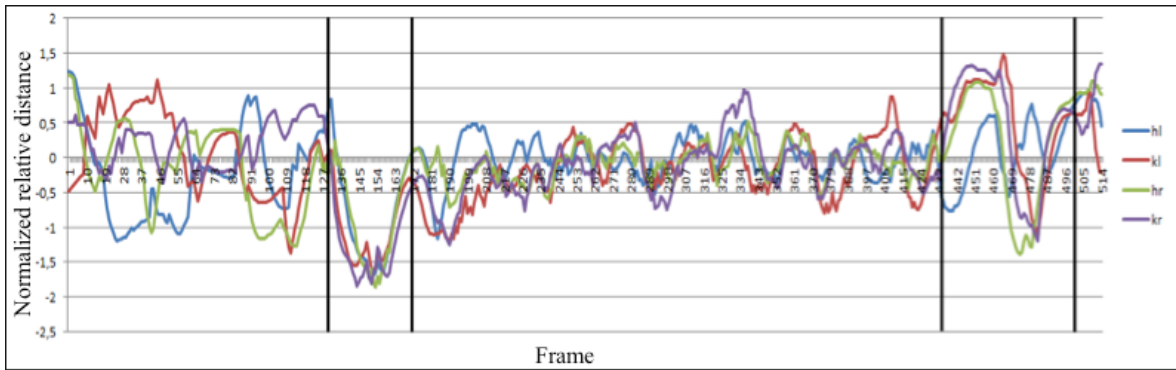


Figure 19 - Relative Distance from the shoulders midpoint to selected joints. Demarked areas between vertical lines are related to Lying down and getting up activities respectively

As it was observed with other activities, Figure 19 suggests patterns that can be visually identified with the lying downs and getting up activities suggesting they are prone to be analysed, characterized and used to label the detected activities. This in conjunction with our detection solution has the potential a full monitoring solution as depicted in Figure 18.

Another important evolution in the future is to provide a full IT support namely to store, query and classify the data namely through a web portal. A web portal with all the information captured is feature that can be implemented in the future. This portal could also be complemented with a mobile application that can convey the information gathered and also be a platform for the warnings to be given out.

8. References

- [1] Horn, L. B. “Reducing the Risk of Falls in the Elderly.” *Rehab Management* 96 (2000): 36–38.
- [2] Gjoreski, H., M. Lustrek, and M. Gams. “Accelerometer Placement for Posture Recognition and Fall Detection.” In *Intelligent Environments (IE), 2011 7th International Conference On*, 47–54, 2011.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6063364.
- [3] Tolkiehn, M., L. Atallah, B. Lo, and G. Z. Yang. “Direction Sensitive Fall Detection Using a Triaxial Accelerometer and a Barometric Pressure Sensor.” In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, 369–372, 2011. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6090120.
- [4] Rougier, C., E. Auvinet, J. Rousseau, M. Mignotte, and J. Meunier. “Fall Detection from Depth Map Video Sequences.” *Toward Useful Services for Elderly and People with Disabilities* (2011): 121–128.
- [5] Bao, L., and S. Intille. “Activity Recognition from User-annotated Acceleration Data.” *Pervasive Computing* (2004): 1–17.
- [6] Zhang, S., P. McCullagh, C. Nugent, and H. Zheng. “A Theoretic Algorithm for Fall and Motionless Detection.” In *Pervasive Computing Technologies for Healthcare, 2009. PervasiveHealth 2009. 3rd International Conference On*, 1–6, 2009.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5191216.
- [7] Stone, E. E., and M. Skubic. “Evaluation of an Inexpensive Depth Camera for Passive In-home Fall Risk Assessment.” In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference On*, 71–77, 2011.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6038771.
- [8] Fujiyoshi, H., and A. J. Lipton. “Real-time Human Motion Analysis by Image Skeletonization.” In *Applications of Computer Vision, 1998. WACV’98. Proceedings.*,

-
- Fourth IEEE Workshop On*, 15–21, 1998.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=732852.
- [9] Chun, S., K. Hong, and K. Jung. “3D Star Skeleton for Fast Human Posture Representation.” *Proceeding of World Academy of Science, Engineering and Technology. Auckland: WASET* (2008): 273–282.
- [10] Auvinet, E., F. Multon, A. Saint-Arnaud, J. Rousseau, and J. Meunier. “Fall Detection with Multiple Cameras: An Occlusion-resistant Method Based on 3D Silhouette Vertical Distribution.” *Information Technology in Biomedicine, IEEE Transactions On*, no. 99 (2010): 1–1.
- [11] Foughi, H., B. S. Aski, and H. Pourreza. “Intelligent Video Surveillance for Monitoring Fall Detection of Elderly in Home Environments.” In *Computer and Information Technology, 2008. ICCIT 2008. 11th International Conference On*, 219–224, 2008. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4803020.
- [12] Rougier, C., and J. Meunier. “Demo: Fall Detection Using 3D Head Trajectory Extracted from a Single Camera Video Sequence.” *Journal of Telemedicine and Telecare* 11, no. 4 (2005). http://www.computer-vision.org/4security/pdf/montreal-fall_detection.pdf.
- [13] Boulay, B., F. Bremond, M. Thonnat, and others. “Human Posture Recognition in Video Sequence” (2003). <http://hal.inria.fr/inria-00494249/>.
- [14] Hu, C., Q. Yu, Y. Li, and S. Ma. “Extraction of Parametric Human Model for Posture Recognition Using Genetic Algorithm.” In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference On*, 518–523, 2000. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=840683.
- [15] Fu, Z., E. Culurciello, P. Lichtsteiner, and T. Delbruck. “Fall Detection Using an Address-event Temporal Contrast Vision Sensor.” In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium On*, 424–427, 2008. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4541445.