Automatic Performance Analysis in Trampoline from RGB-Depth Data

Carlos Puig Toledo

Abstract

In this master thesis, it is proposed to capture multi-modal RGB-Depth data obtained by a Kinect device, synchronize and align the captured modalities with a frame rate near 30FPS, and use Computer Vision techniques and methods as homographies in 2D or planar segmentation and euclidean clustering extraction in 3D, in order to extract a relevant indicator as is the landing point concerning a jump. The problem is divided in two parts related to their dimensionality. On one part we focused in estimate a homography to transform the bi-dimensional image of the trampoline in order to undo the perspective view. The other part consists in the usage of the depth information to reconstruct a 3D model of the scenario and estimate in it the jump location in the mesh.

Author: Carlos Puig Toledo, carlos.puig@e-campus.uab.cat
Advisor 1: Sergio Escalera, HuPBA, Universitat de Barcelona and Computer Vision Center
Advisor 2: Jordi Gonzalez, Computer Science Department, Universitat Autonoma de Barcelona and Computer Vision Center
Advisor 3: Xavier Baro, Universitat Oberta de Catalunya and Computer Vision Center
Advisor 4: Albert Clapes, Universitat de Barcelona and Computer Vision Center
Thesis dissertation submitted: September 2014

ACKNOWLEDGMENT

We would like to thank Josep Escoda and Xavier Balius from CAR and Ricard Jimenez from ASCAMM from their support during the development of the project.

Index Terms

CONTENTS

LIST OF FIGURES

I. Introduction

REGARDING the field of computer vision, methods, theories and systems have been developed to perceive and interpret the movements in different videos of different scenarios. The purpose is to acquire this movement which for us is clear as can be: know which are the different objects or people, where they come from and where they go, know their behavior and deduce their purposes.

The evaluation of the movement in videos or image sequences implies different tasks as: acquisition, detection, tracking, recognition, and behavioral study. In many computer vision applications, the most common approach is to segment the objects and their background surfaces which we can be interested or they can be discarded. This allows to lead with cases where the objects do not have always the same shape or it is not well defined, as in this case where are people moving cause they are doing sport activities.

Thankfully, there is a huge amount of different techniques in order to segment objects and people through a Kinect camera. The big point of this camera is that it allows to acquire images in RGB-Depth format and with this depth information the scenario can be recreated in a 3D world, making easier to know what parts are more distant and more nearby, which are the surfaces, etc.

In this master thesis in collaboration with CAR Sant Cugat, it is proposed to capture RGB-D data with a Kinect device, which has great advantages.
On the one hand, it allows the acquisition of video about 30FPS, which in most of the sports and particularly at certain times of the exercise is enough. In addition, the error rate in general does not exceed 5-10 cm in the depth map, which is sufficiently accurate in this case, which is to know at what point the athlete touches the mesh trampoline.
On the other hand, many sports are performed outside of a controlled environment, such as a laboratory where the experiments would be conducted. Many sports are performed in open places with lighting changes, etc. which results in a particular setup of cameras and not optimal because it may have to be changed afterwards and may involve great expense. With Kinect this problem is much less, since it is just one camera that can be placed on a simple structure such as a tripod to be easily transportable, can record indoor and limited outdoor scenarios (only by a range and depending the sun-light cause interrupts the IR sensor), and adjust the parameters to suit eg lighting is simple to configure cause is a single camera and not a set of them.

Among the problem in measuring the performance of athletes jumping on a trampoline, we have focused on solving two sufficiently specific parts of the problem with Kinect. One reason is to divide the problem in easier parts and also there are already implemented many methods and techniques that help us to solve each part, that are: segment the athlete from the rest thanks to the clustering of objects in 3D and to detect the flat surfaces as in this case is the trampoline.

Furthermore, there is another point to be solved. Since the camera is located at the side of the trampoline, the sequence is filmed in perspective. This perspective must be transformed by a homography in 2D to build a model of the mesh where we can see the results. This step is necessary because there are different sizes of trampoline depending on the type, age and weight of athletes, so we can not take a single model for all cases.

Fig. 1: Olympic exercise on a trampoline

A. Motivation

The motivation for this project is the constant need for athletes, regardless of sport, to improve its performance. With technology advancing every day, we are able to create systems that help people to notice details and evaluate physical activity that helps them to be better. Thanks to this, we are also able to help people evaluate and qualify athletes exercises, such as referees and judges. Some examples of technology applied to sports are: the hawk-eye that helps the referees to validate or reject some plays in rugby, tennis and soccer among others; or underwater cameras that help judges to whistle fouls in waterpolo or rate the exercises in synchronized swimming.

We can see how in this big and varied world as is sports, whether collective or individual, the technology is very present and in a specialty as is the trampoline, which is Olympic since 2000, it is necessary to apply it. On the trampoline already different technologies exist that help, both judges and coaches, measure the performance of athletes, such as the use of high speed cameras for viewing more detailed exercises or using a laser at the level of the mesh to compute the flight time at each hop, which is directly related to the score attributed to each athlete.

Therefore in this project, we thought of other ways of measuring the exercises, and other parameter that is linked directly to the score is the point where the athlete touches the mesh. This is cause, in the mesh we find a rectangle in which athletes must land at each hop. If the contact point is outside the rectangle they can be penalized with a drop in the overall score of the exercise. However, the limitation of this rectangle is not entirely strict, i.e., the athlete can go outside the rectangle as long as it is not excessive (almost touching the edge of the mesh). The following section describes in more detail this sport and how are this actions penalized.

B. The trampoline sport

The trampoline (or also called trampolining) [2], [3], [5] is a competitive Olympic gymnastic sport in which gymnasts perform acrobatics while bouncing on a trampoline. These can include different combinations of jumps to make the exercise more complex with forward and backward somersaults, twists, etc.

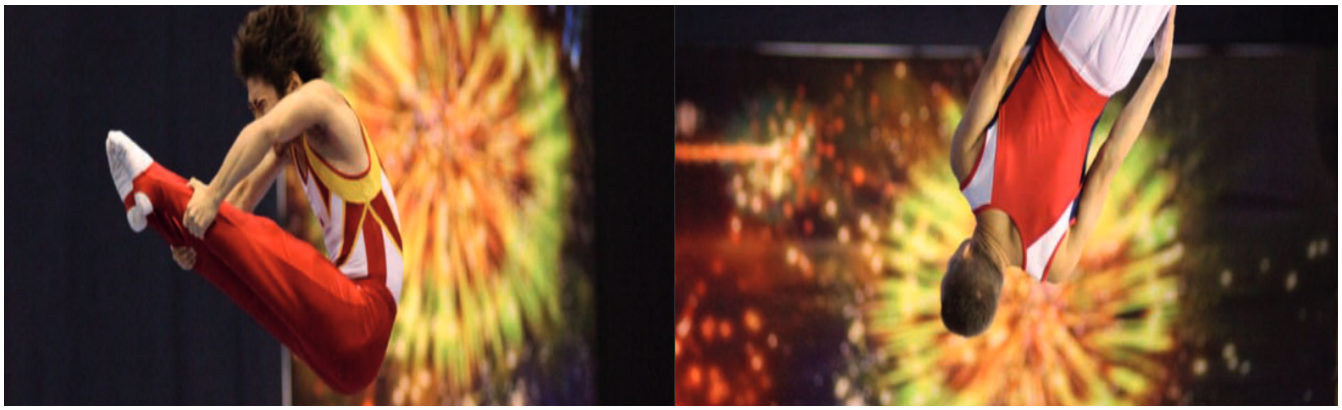


Fig. 2: Athletes performing an exercise

The first concepts date back to 1890s where Billy Bouncer and John Hayes observed trapeze artists performing tricks while bouncing on the safety net or a blanket like the firemen used to rescue people that needed to jump off the buildings. This techniques was used to entertain audiences but this was the beginning of a new sport. The first modern trampoline was built by George Nissen and Larry Griswold in 1934 at the University of Iowa. It was initially used to train tumblers and astronauts and as a training tool to develop and hone acrobatic skills for other sports such as diving, gymnastics and freestyle skiing. People enjoyed the sensation so much, they began to trampoline for sheer fun, and it became popular in its own right. Trampolining made its first appearance at the 2000 Games in Sydney, with men's and women's competitions. The number of events (two) has remained unchanged since then.

The frame of a competitive trampoline is made of steel and can be made to fold up for transportation to competition venues. The trampoline bed [4] is rectangular 4.26 by 2.13 meters in size fitted into the 5.2 by 2.13 meters frame with around 90 steel

springs. The bed is made of a strong fabric, although this is not itself elastic; the elasticity is provided only by the springs. The size of the trampoline, the materials, or the number of steel springs may vary depending of the usage (school, training of other sports, recreational, etc.), and the age and weight of the athletes.

(a) Trampoline structure and sizes

(b) Trampoline springs

Fig. 3: Trampoline elements

There are three related competitive rebound sports, synchronized trampoline, tumbling and double mini-trampoline. Due this project consists in evaluate only an athlete in one trampoline there is provided the explanation and rules for the individual trampoline.

Individual Trampoline consists of an individual competitor performing two routines on the trampoline. These routines consist of a 1st voluntary (compulsory elements) and a 2nd voluntary routine (optional).

In most competitions, skills are performed from ten contacts with the bed, starting the routine and ending the routine on the feet. Skills range from aerial shapes (tucks, pike and straight bounces) to multiple somersaults with twists. Skills receive difficulty points according to body position, the degrees of rotation and twist executed. The 2nd voluntary routine also consists of a combination of ten consecutive different skills. The competitor selects the skills. This routine is judged on the performance and a degree of difficulty score is added to this total. Competitors will use a combination of ten different skills that can see the world's best athletes performing a combination of double and triple somersaults with multiple twists, performed at heights of up to 8 meters above the ground, to show-case the aesthetically pleasing and awe inspiring rebounding sports spectacle of trampoline. From January 2011, both routines will be awarded a Time of Flight bonus. Routines will be timed either manually or electronically. The longer the time of flight, time spent in the air, the higher the points bonus awarded to the routine and the time (in seconds) will be added onto the score for that routine.

Some interesting rules [1] in this project consist in those that penalize the execution in relation to landing in some parts of

the bed. This is the list of possible deductions by the chair judges (the deductions are subtracted from the maximum mark):

Deduction	Points
Touching the springs, pads, frame or safety platform	0.6
Landing/falling on the springs, pads, frame or safety platform and spotter mat	0.8
Landing/falling outside the area of the trampoline	1.0

C. Objectives

This project consists to acquire multi-modal images by a Kinect and then use computer vision techniques to get the landing point of the athlete in the mesh. For this the project is separated in four steps to divide the problem in a pipeline.

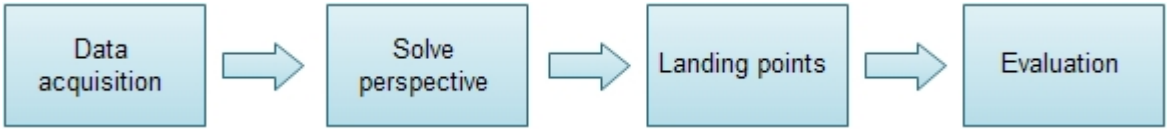


Fig. 4: Pipeline of the implementation of the project and future behaviour of the application

This are the different objectives related to the different steps of the pipeline:

1) Data acquisition: implement the necessary software to get the sequence of multi-modal images from the Kinect device in the major frame rate possible and store it in disk to subsequently process by the different computer vision methods. In this step an important part is required which is the depth map registration. If we take in detail the depth image that Kinect returns, we can realize that it do not corresponds exactly with the RGB image due to the camera and sensor calibrations. For this reason is necessary the registration of the depth map to the RGB image in order to obtain depth and color information simultaneously.

2) Solve the perspective: since the camera is placed on a side of the trampoline we get a perspective view of the mesh. This is a problem because we cannot relate correctly the point in 3D where we estimate the athlete lands to the point in 2D that corresponds in the mesh. To solve this is necessary to find the correct homography in order to transform the image to get an upper view of the trampoline.

3) Landing point: the main task of the project is to estimate the point of the mesh that the athlete touches in every jump. In order to solve this we have to take the depth-color information and create a 3D point cloud to work with. Once the cloud is built we can apply 3D algorithms in it to e.g. segment the athlete, the mesh, the background, etc.

4) Evaluation: finally with the results we compare the estimated points to the real points touched in the mesh in a qualitative way, as is seeing the correspondence in the RGB image.

II. State of the art

A. Scientific documentation

First of all, regarding the scientific documentation from the libraries as papers in journals or workshops, there is a lot of information of multidimensional image representation, 3D vision and reconstruction, etc. In this thesis, it is not a setup of multi-camera model or a kind of reconstruction of the scenario, our purpose is only get the RGB-Depth data to determine when the bed is deformed by an athlete. Thus, an important document is [6] by Luo et al., because although the research was carried out with a pair of stereo cameras, it shows how to measure 3-dimensional deformations in deformable and rigid bodies, which is helpful to determine when the athlete impacts the elastic bed.
Another important document is [7] by Dapoto et al., which explains a system to determine and represent 3D trajectories. In

this project may not be an essential part but helpful due to knowing the trajectory of the athlete could be easier to estimate his falling point.
However, these researches are made with stereo cameras to reconstruct the 2D images to a 3D scenario, but this project is performed with a Kinect device. Thus, a certain knowledge of the camera hardware and specific software is needed, aside from the theory mentioned, as is presented in the next sections.

B. Kinect

The Kinect sensor [8] for Xbox 360 was launched in November 2010 as a natural user interface for the Xbox 360 console developed as a gaming interface but has created significant interest in a number of different areas, particularly focused within the robotics community. The Kinect serves as a simple and low cost structured light scanner, packaged into a single unit, thereby opening up the potential for a large number of 3D vision applications.
It is a horizontal bar connected to a small base with a motorized pivot that can be controlled from application. The device features an RGB camera, depth sensor and multi-array microphone, which provide with the default proprietary software full-body 3D motion capture, facial recognition and voice recognition capabilities. The depth sensor consists of an infrared laser projector combined with a monochrome CMOS sensor, which captures video data in 3D under any ambient light conditions. The sensing range of the depth sensor is adjustable.

The default RGB video stream uses as maximum 8-bit VGA resolution (640x480) with a frame rate of 30FPS (but the hardware is capable of bigger resolutions at lower frame rate). The monochrome depth sensing video stream is in VGA resolution with 11-bit depth, which provides 2048 levels of sensitivity and its range is 1.2-3.5 meters, limiting its use to indoors and certain spatial conditions and purposes in outdoors.

One of the problems with conventional structured light scanners are very slow to obtain scans as they require light stripes to be projected onto the scene in different orientations in order to obtain the 3D geometry of the scene.
Although still strictly speaking a structured light scanner, the Kinect works in a very different manner. Instead of a series of different stripes the Kinect uses a speckle pattern of dots that are projected onto a scene by means of an IR projector, and detected by an IR camera, as shown below.

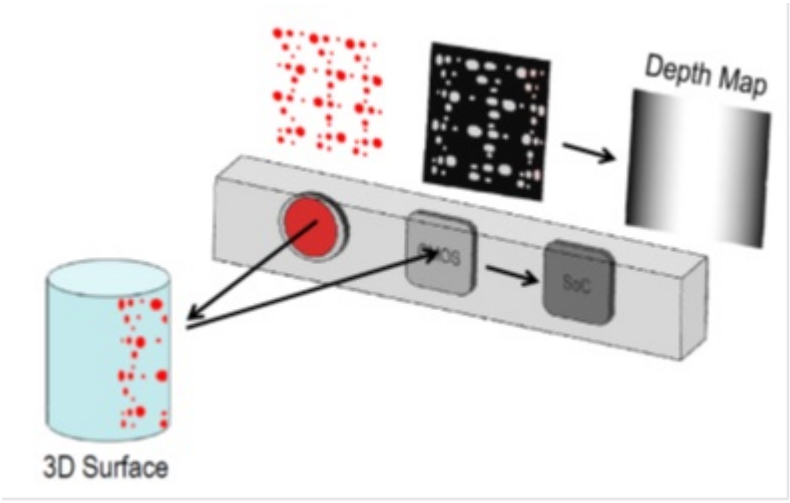


Fig. 5: Kinect system overview

Hard coded into the Kinect at manufacture is a reference pattern of the speckle pattern. Each IR dot in the speckle pattern has a unique surrounding area and therefore allows each dot to be easily identified when projected onto a scene. The processing performed in the Kinect in order to calculate depth is essentially a stereo vision computation. The mathematical algorithm picks a particular dot in the reference pattern and then looks for that dot in the observed scene by also looking for its eight unique surrounding pixels. Once found in the scene its disparity can be determined and used in conjunction with the focal length of the IR camera used to detect the speckle pattern and the baseline between the projector and the camera in order to determine the depth of that given point in the scene. This process is then simply repeated for each point in the reference pattern.

The IR speckles projected by the Kinect are of three different sizes that are optimized for use in different depth ranges, meaning that the Kinect can operate between approximately 1m and 8m. The IR source is a constant source, un-modulated emitter that is directed at a diffractive optic element (DOE), as shown below in this figure.

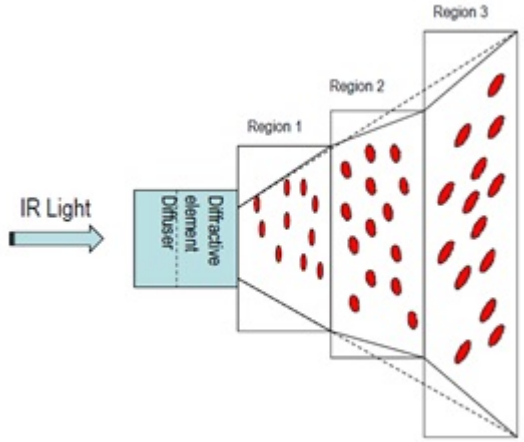


Fig. 6: The Kinect's DOE and associated speckle output

As part of the processing the Kinect initially identifies which range area a particular object lies within and then only the speckles optimized for that particular range will be used for calculating depth in that area. The ability to project different sized dots in one go is one of the considerable advantages of the Kinect and one of the ways in which it is able to operate over such a wide range with a single scan.

One of the clever parts of the Kinect is that in addition to pure pixel shift, it also compares the observed size of a particular dot with the original size in the reference pattern.

C. Kinect in sports

In this section some examples of using Kinect in the world of sports are shown, not only in sports activities such as training or competition but also in rehabilitation. These review arise both research groups and companies dedicated to applying technology in the sports scope and they are a great reference since our project involves also a center.

In the research project called Depth Biomechanics in [9] we can find an analysis of player segmentation and tracking from the environment using depth cameras instead of color/intensity information. The objective of this is to distinguish the badminton player to track his position in the court first using known trajectories and locations (marked on the floor) and then the estimated position using depth cameras are compared using a three-dimensional motion capture system.
This work is very interesting in our project, cause in our case, we also have to segment and track a person and estimate its location.

Another line of investigation in the project of Depth Biomechanics [10] was the study of the potentially uses of Kinect in coaching and education for this three application areas: segment tracking, segment/body scanning and blob tracking.
For this task they used a mannequin that was scanned with the Kinect and a laser scanner. The geometries were truncated to create torso segments and compared. Separate shoulder abduction (100 to 50) and flexion motions (0100) were recorded by the Kinect (using free and commercial software) and a Motion Analysis Corporation (MAC) system, and then the segment angles were compared.
One of their conclusions is very relevant in this project, and this is because they say that the Kinect's low cost and depth camera are an advantage for sports biomechanics and motion analysis, although segment tracking accuracy is low.

Another example is the work by Holsti et al. [11] which is very related to this project cause they describes their effort in developing trampoline training games using computer vision technology. The study is part of a project about developing digitally augmented exercise environments for faster, safer and more engaging sports training. They also test the prototype with circus students and people with no background in trampolining which is interesting but in our case will be only athletes

with certain acknowledge in the sport.

The article by Yan Long Che et al. [12], is based on motion capture device from Kinect, combined with the feature of sports activities, to provide effective solutions for sports training, so as to enhance the level of sport athletes. As in our project they apply the motion capture technology of the Kinect in order to improve the performance of the athletes. They focused in the problem that the traditional sports training coach is one-to-one or one-to-many face-to-face training, it not only take the limitation of time and place, but also the high cost of training, coach resources nervous. For this they hope professional sports training or national training began to apply motion capture to sports training.

Also related with sports but in the field of rehabilitation we can find the study by Fernandez-Baena et al. [13] which is a comparison of the precision in the computation of joint angles between Kinect and an optical motion capture professional system. They obtain a range of disparity that guaranties enough precision for most of the clinical rehabilitation treatments prescribed nowadays for patients. Although this study is related to rehabilitation is also an important field in sports, and the analysis of the Kinect precision in motion capture help us in order to segment the athlete and even more in the future work where probably the pose and angles of the professional will be a case of study.

As the previous paper, Chuan-Jun et al. [14] in their article also relate the Kinect with the rehabilitation. In this case they present a development of a Kinect-based system for ensuring home-based rehabilitation (KHRD) using Dynamic Time Warping (DTW) algorithm and fuzzy logic. Their objective with this is to offer assistance for patients to conduct home-based rehabilitation without the presence of a physician and to avoid adverse events which differs from the previous study where the rehabilitation treatment was think to be done in a clinic with supervision.

On the other hand we find Experimedia [15], an innovator company in developing and implementing ideas in the field of sports thanks to new technologies such as cameras and sensors. As a result they have managed to make several projects in important sports centers, such as the same where this thesis focuses (CAR Sant Cugat). Some of their projects that can serve as reference for this thesis can be: iCaCoT (an interactive camera-based coaching and training for visitors and sports enthusiasts) or 3D Media In Sports (investigates the usefulness of 3D information in high performance sports training centers).

D. Point Cloud Library (PCL)

Fig. 7: Point Cloud Library logo

In order to research with the Kinect data probably the most powerful library is PCL [16] which allows getting the images from the Kinect sensor and also processing them. The Point Cloud Library is a large scale, open project for 2D/3D image and point cloud processing. The PCL framework contains numerous state-of-the art algorithms including filtering, feature estimation, surface reconstruction, registration, model fitting and segmentation. These algorithms can be used, for example, to filter outliers from noisy data, stitch 3D point clouds together, segment relevant parts of a scene, extract keypoints and compute descriptors to recognize objects in the world based on their geometric appearance, and create surfaces from point clouds and visualize them.

PCL is open-source cross-platform implemented in C++ as a template library, and has been successfully compiled and deployed on Linux, MacOS, Windows, and Android/iOS. This modularity is important for distributing PCL on platforms with reduced computational or size constraints.

III. METHOD

A. Data acquisition

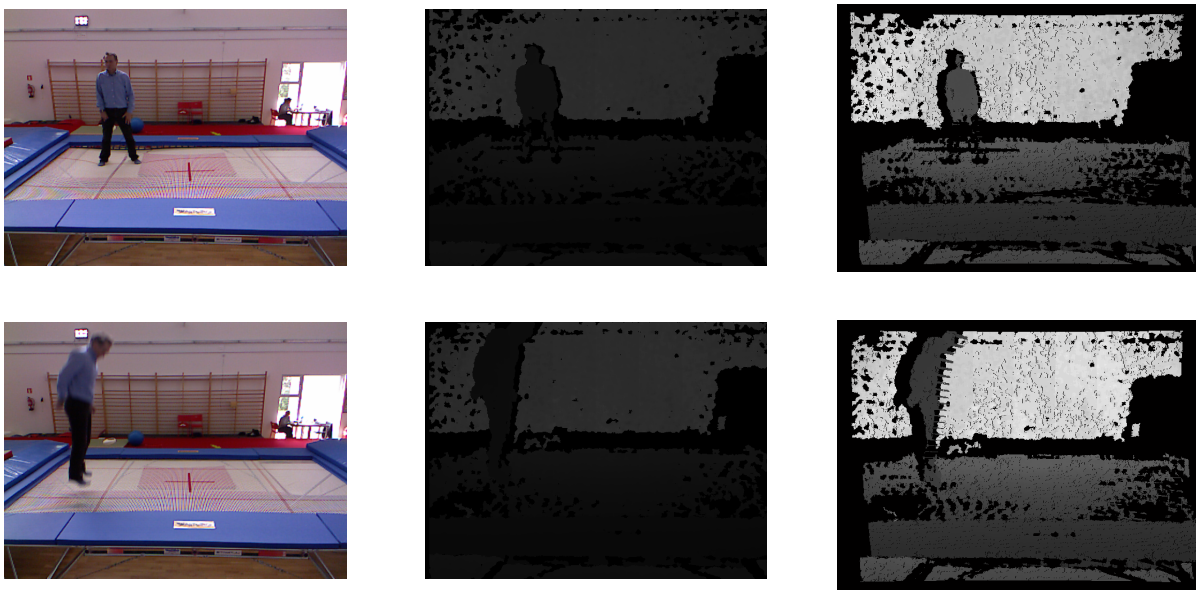
The first step is to obtain the RGB-D images by the Kinect in the trampoline scenario while an athlete is performing an exercise. For this purpose is needed to implement simple software capable to get the images from the device and store correctly

in the disk. This is done with the help of the Windows Kinect SDK [17] that provides a great amount of functions and methods not only to acquire the images, but also to perform body tracking and recognition, face recognition, etc. Although the SDK of Microsoft is not open-source code, which means that we cannot change the inherent functions that they provide for the device, we only need the simple functions that allows us to the get and store the images and not the other functions that performs more complex algorithms. This is a point to take in account cause if in the future we also want to estimate e.g. the athlete skeleton or his pose, the SDK provides this tools but we are not allowed to change the code of the algorithms, and probably will be necessary a migration of the acquisition system to another platform as OpenNI that is open-source.

Once we get the software ready is time to go the trampoline scenario and record. Unfortunately by schedule combination issues the record made was not with a professional athlete and one of the advisors in the sports center offered himself to jump in the trampoline. This was a problem cause the scenario changed a little because the athletes during the exercise do not jump in the same way. For example, in our sequence can be seen that the person sometimes jumps with legs apart, facing the camera or even supporting the arse, when in reality an athlete would fall with feet close together and side to the camera, never in front.

Another problem was that the software made in order to acquire the images was running in debug mode and in a computer quite old. Thus, the ability to save the images to disk was with a frame rate of 15 20FPS which are not the maximum performance that we can from the Kinect (near 30FPS). To solve this problem there are different possible solutions concerning both hardware/software.
On one hand, the application in the future will run in release mode making the processing faster and probably we only have to save the depth map and not the color image cause there will be no need to compare with the color sequence. On the other hand, the critical time consuming is the writing to disk. This is strictly related to the specification of the hard drive disk and other hardware components. One way to improve it would be to use a solid state disk that reduces a lot the time of writing.

As is explained in the objectives section, there is an important step in the acquisition which is the registration of the depth to the color information. Due the calibration of the color camera and the depth sensor, the pixels are not related in the two images and a technique to combine each pixel in the depth map with a color of the RGB image is needed. Thus allows us to build a 3D reconstruction with the correct color in every point.



(a) Color image

(b) Depth map

(c) Depth map aligned and contrast adjusted only to visualize it

Fig. 8: Data acquired

B. The perspective problem

Since the camera is located on the side of the trampoline, it is viewed from a side perspective. This fact is easily visible through the mesh lines, which can distinguish easily a couple of them that are not parallel between them but tend to a vanishing

point. Luckily when we take the data that fact already had been taken into account and the camera was located, the best we could, parallel to a side (although you will always find a small error in calculating the homography that the parallel is not perfect), otherwise for example, if it had been located on a corner, the perspective would affect the two pairs of lines that we wanted to be parallel. So the basic idea was that the two pairs of lines that delimit the central rectangle of the mesh, two horizontal and two vertical, would be parallel to each other, and had into account the previous fact we found that the horizontal lines already accomplish this requirement.

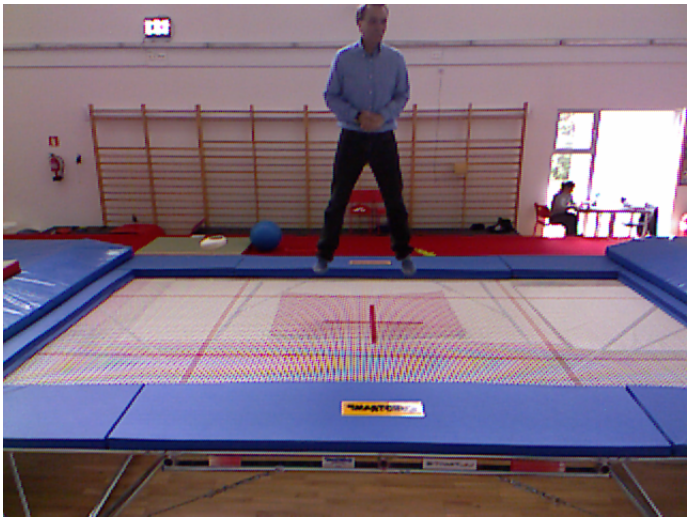


Fig. 9: Color image selected to estimate the homography due to the athlete do not occlude the mesh

One of the first ideas was to segment the red lines painted on the mesh after estimating a homography that would make them parallel. However, information obtained by segmenting color red lines was a task too complex as there was a small noise because the mesh has holes and creates an effect similar to aliasing in the Kinect RGB camera, and also through the holes we could see the red floor, the red lines are quite thin and the their color varies according to the area once it is worn.

Therefore, this option was discarded, and the next idea was to segment the edge of the mesh, i.e. the blue area surrounding the mesh. This part has more uniform color and texture, as it is a smooth surface. Taking the data from these areas in the HSV color space, as it is more robust to changes in illumination and intensity, was performed a GMM to find the optimal threshold in the Hue channel to segment the blue mat.

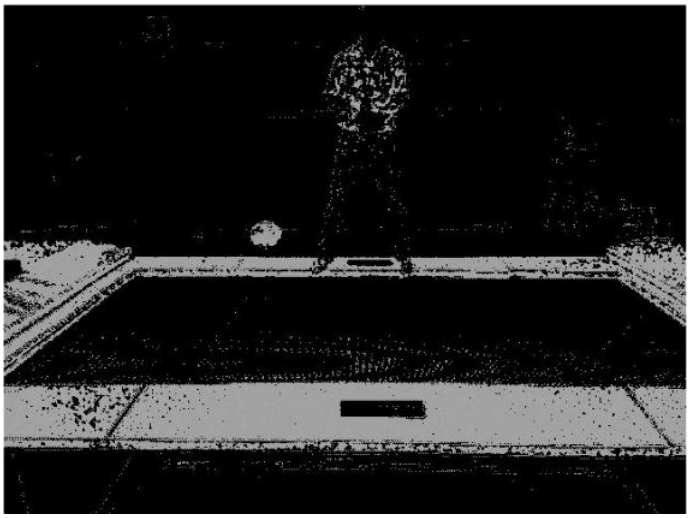


Fig. 10: Segmented parts by a threshold in Hue channel

Once we had distinguished much of the area, various combinations of morphological operations as opening and closing were

applied to make more robust the area estimated by segmentation. We also knew that the mesh was inside this area founded and with a flood-fill algorithm we segmented this inner area.

Fig. 11: Segmented mesh

Thanks to this, we managed to find the edges of the segmented mesh which are parallel to the red lines that we wanted to find at first. However there is a fact to consider, taking as reference the edges of the mesh instead of the interior lines, when estimating the homography, we would get a small error.

To segment edges we used Canny as edge detector and then as a feature extraction technique we used the Hough transform in order to detect the lines. To make the algorithm more robust Hough various parameters and constraints were specified separately to find the horizontal and vertical edges.

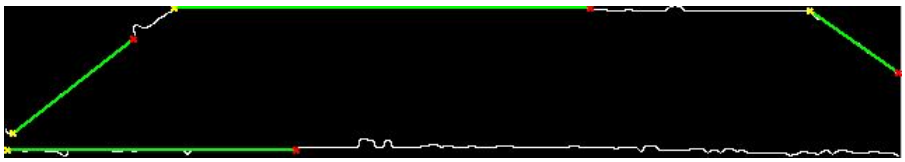


Fig. 12: Lines founded by Hough transformation

Having identified the candidate lines that represent the edges of the mesh, we chose those that we found closer to the center of the mesh because since we would have a lower error. With these four lines in vector form we found the points where they intersect thanks to the vector cross product. Then with these four points we estimated the homography based on the idea that those points that belong to the corners of the mesh, must had a rectangular shape. Thus, we performed a mapping to a rectangle described by us to define the new view of the mesh (as to be seen from above).

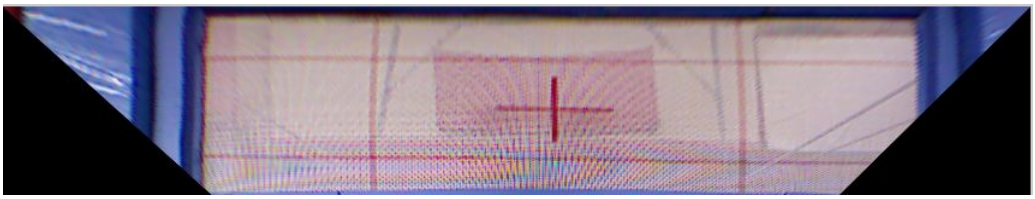


Fig. 13: Result of apply the estimated homography to the mesh

C. Finding the landing points

This project part was all implemented with the help of the PCL libraries which also include the Visualization Toolkit VTK that allowed us to visualize our work in an easy way taking in account is a 3D visualization.

The first step was to read the depth images recorded with the Kinect and make a point cloud. For this purpose we implemented a function that reads the image and for each pixel get its value which represents the distance from the camera to this point in the 3D world. However this was not enough cause we also had to compute the x and y coordinate in the world coordinates. First we had to compute the inverse focal which depends on the intrinsic parameters of the Kinect that were known.

$$inversefocal = \frac{\frac{1}{285.63}}{\frac{imagewidth}{320}} \tag{1}$$

Once we had computed the inverse focal we had to transform the three coordinates to real world coordinates. Obviously the z is already in the real world coordinate.

$$realworldX = (x - imagewidth) \cdot invfocal \cdot z \tag{2}$$

$$realworldY = (y - imageheight) \cdot invfocal \cdot z \tag{3}$$

$$realworldZ = z \tag{4}$$

Then with this new coordinates we constructed a point cloud where each point relates to a pixel. We copied also the color information.

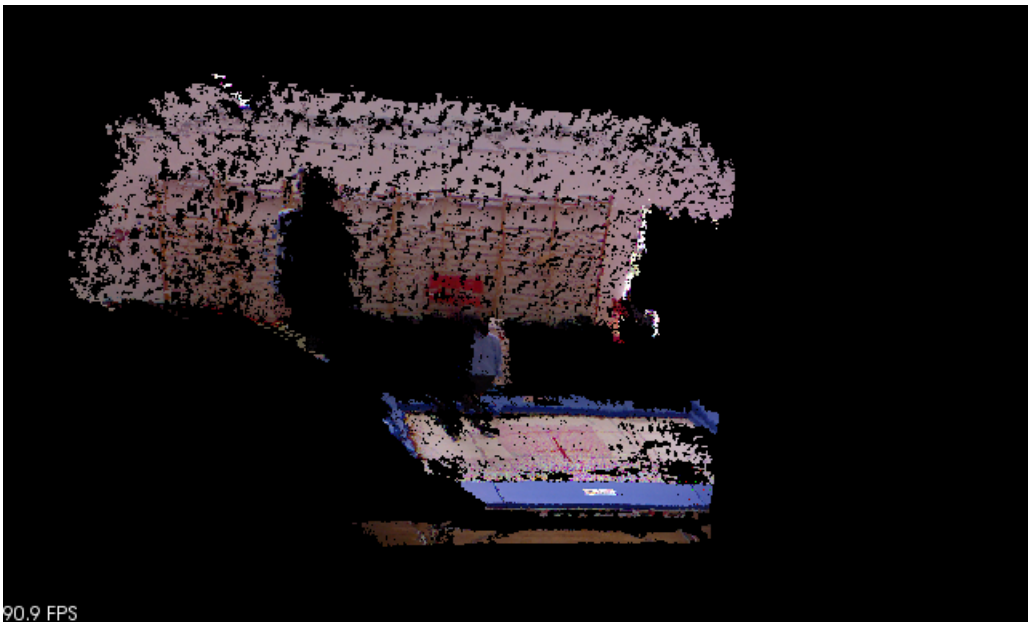


Fig. 14: Reconstruction of the depth map in the 3D world

Once we had the point cloud we had to down-sample it by a grid voxel filter of 1 cm due to the huge amount of points caused the rest of algorithms and methods that were applied ran very slow or they even finished.

With that point cloud we performed the same steps as when we wanted to estimate the homography in order to undo the 2D perspective, i.e., we first sought the blue area of the trampoline and then the inner part was segmented, which belongs to the mesh that was the interesting.

Then with the point cloud of the segmented mesh, we applied a planar segmentation algorithm, not to detect the planar surfaces cause we had the mesh totally segmented an with anyone jumping on it, but to get the estimated plane parameters. This method apart of return the set of points that belongs to a planar surface, it also returns the estimated plane parameters of this surface in ax + by + cz + d = 0 form.

With this what we wanted was to compute the normal to the plane in order to find a 3D transformation that translates the

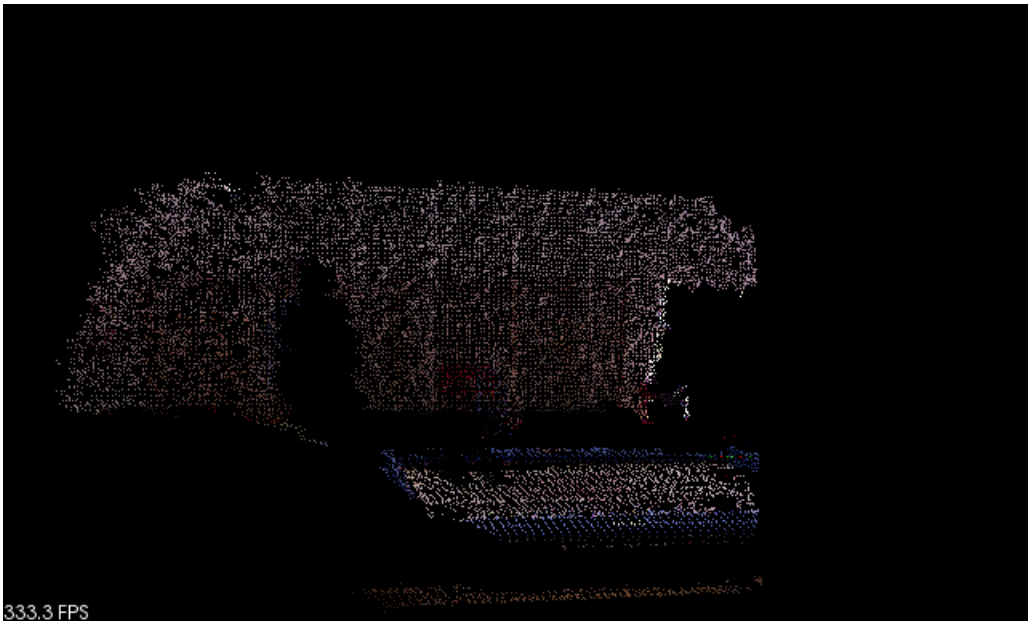


Fig. 15: Downsampled point cloud by a grid voxel filter of 1cm

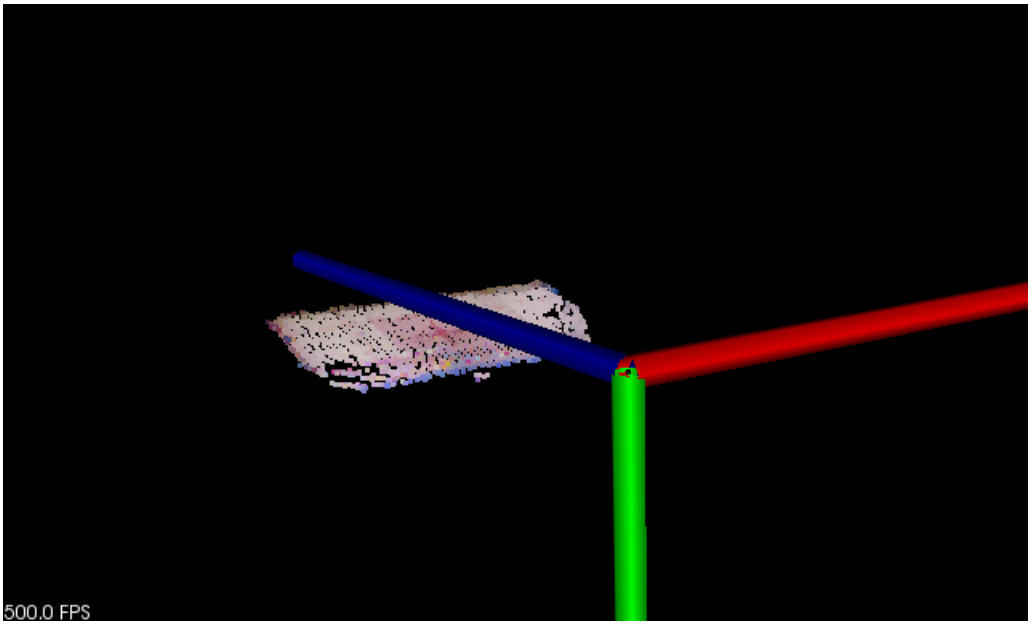


Fig. 16: Segmented mesh in 3D

plane of the mesh to the plane formed by the x and z axis. The idea was if we placed the mesh in the plane formed by those axes the movement of the athlete would be along the y axis and the later computation would be easier. With the plane parameters explained before, the normal vector to the plane is defined by (a, b, c). With this information we could find the angle of rotation to match the normal to the y axis by computing the inner product of the two vectors.
We did an assumption to make the problem easier. Being the camera located on the side of the bed and pointing to the center of the mesh, is only necessary a rotation in the x-axis to match the x and z plane, however, is important keep into account that the parallelism of the camera with the trampoline is not perfect and therefore would require some more rotation in another axis to match perfectly the x-z plane.

$$\Theta = \arccos(\frac{x \cdot y}{|x||y|}) \tag{5}$$

However if we did a rotation in the 3D world all the objects in it would rotate respect to the center of the world. To rotate the mesh from itself first we had to translate it to the origin. To this task we computed the 3D centroid of the mesh plane to get its coordinates. Then we had only to apply a translation inverse of the coordinates obtained.

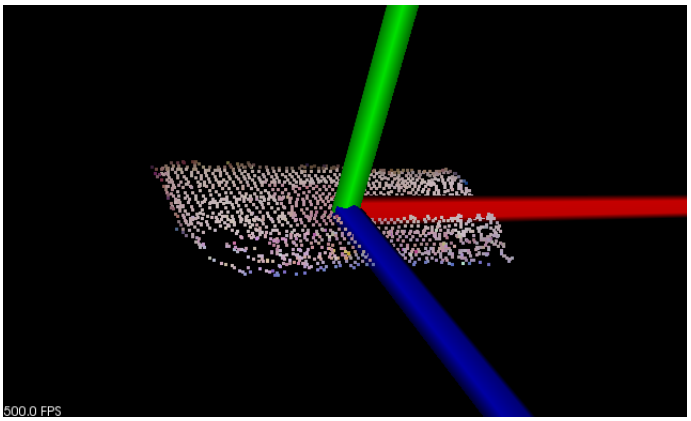


Fig. 17: Mesh translated to the origin and matching the x and z axis

With the mesh correctly positioned the next step was to define a filter that include all area of the mesh in order to segment only the points that are inside when we get a new image of the sequence. To do this we create a cropbox filter that only needs to be specified by a minimum and maximum point at each axis, which were the points at the limits of the segmented mesh.

Having the transformation matrix and the box filter we could, for each image frame of the sequence, segment the points that were in the area of the mesh. The idea was that, when the mesh is in calm the area of the box is filled as a plane structure, but when the athlete hit the mesh the surrounding points of the contact point disappear because they go out of the box pushed by the athlete who in this moment is passing through the box.

The final part was to determine which points belong to the section where the athlete is passing through and when he touches the mesh. By experience we could see that when an athlete hits the mesh, it is segmented quite clearly into two set of points, one of the person and the others the rest of the mesh, and also that the set of points that belong to the person were almost always very low respect to the overall. On the other hand, we could see that the amount of points in the cloud descend considerably when the athlete push the mesh about 20cm or more (which is not a problem because all the exercise jumps sink the mesh more than this).

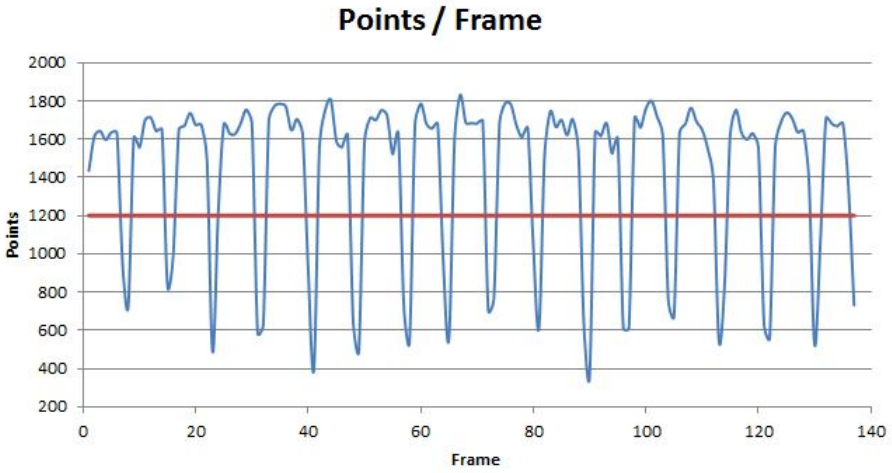
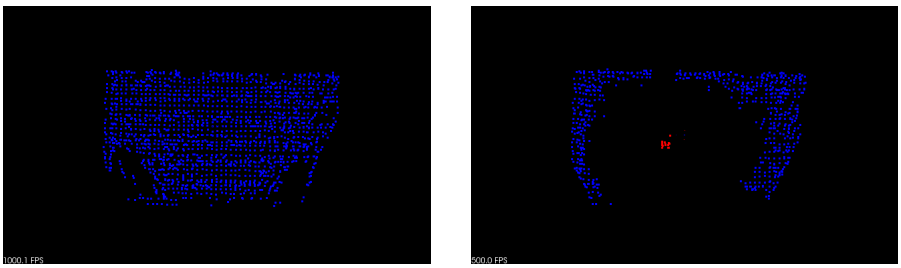


Fig. 18: Number of points in the cloud at each frame and threshold placed in 1200 points

Thus, for when the athlete hits the mesh we defined a threshold in the amount of points in the cloud to determine it. And for where the athlete hits the mesh we used an euclidean cluster extraction method. This algorithm works starting from a point

and looking for its neighbors, if they are close enough, are marked in the same cluster, if not, they remain waiting to another iteration to form other clusters. Regarding this, the smallest cluster of points would be the one that belong to the athlete and we would know the landing location.



(a) Mesh in calm (b) Mesh pushed by the athlete

Fig. 19: Top view of the crop box of the mesh

To know where the location point in the previously computed 2D mesh without perspective is, we implement a mapping equation to pass from the 3D detected cluster in the cloud to the 2D image. First of all, the mesh cloud was placed in the origin and as consequence it had negative coordinates. We had to translate it to positive coordinates to find which was the size of the cloud in x and y. This is computed by:

$$cloudSizeCoordinates = maxPointCloud + |minPointCloud| \tag{6}$$

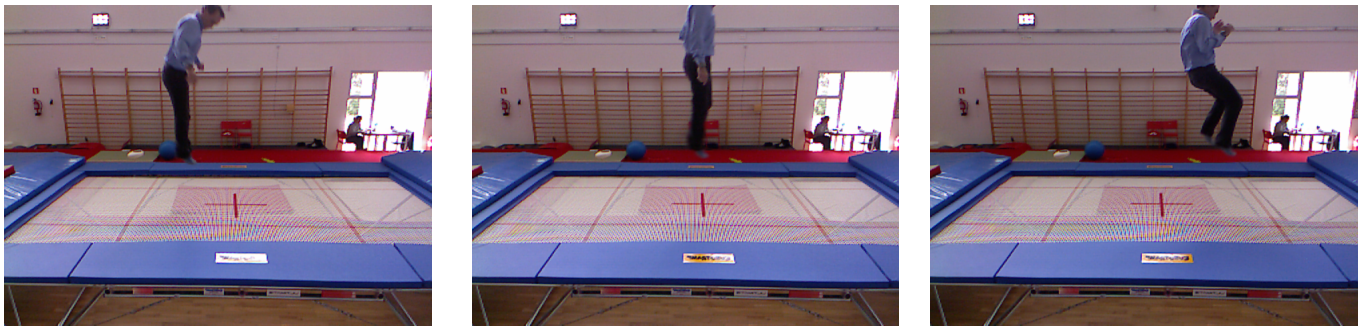
Then knowing the size of the 2D mesh, we defined an equation to directly transform the point coordinate found in 3D to the position in the image, where (a,b) is the location in x and y coordinates in the 3D world and (a',b') is the location in the 2D image.

$$a' = (a \cdot imageWidth/cloudSizeCoordinatesX \tag{7}$$

$$b' = (b \cdot imageHeight/cloudSizeCoordinatesY \tag{8}$$

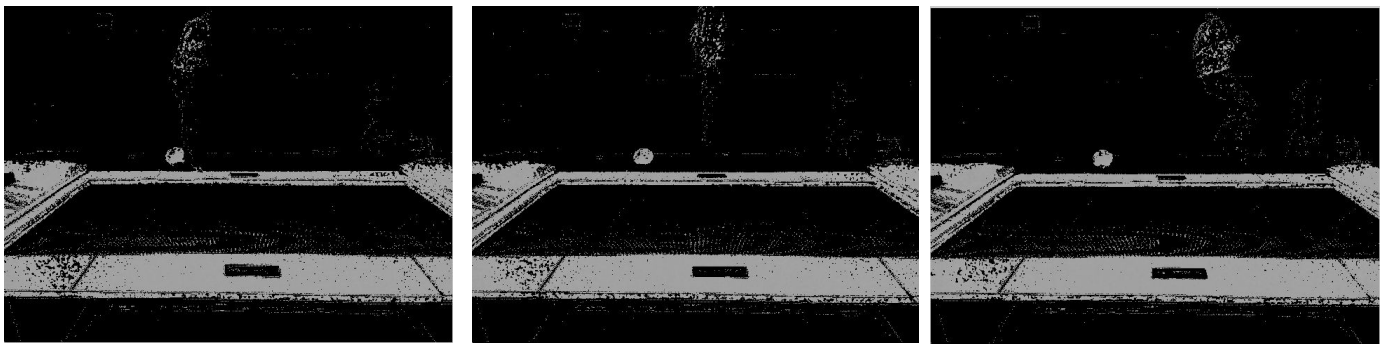
IV. Experiments

For solving the perspective problem and estimate the homography we experimented with three different images where the person do not occlude the mesh.



(a) (b) (c)

Fig. 20: Test images



(a) (b) (c)

Fig. 21: Blue color segmentation in HSV color space

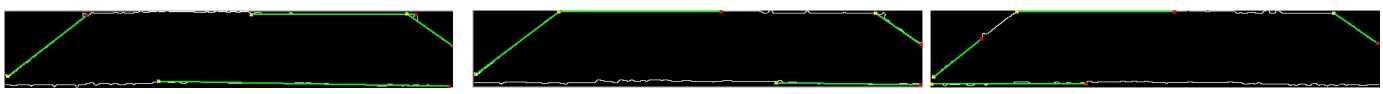
The next step was segment the blue mat. With GMM we found that the optimal threshold value was between 0.6 and 0.65 in a hue value range of 0 to 1.

To make more robust the area segmented various morphological operations were applied. The strategy was to apply closing and opening operations, one after the other and each time with a bigger structure element of square shape (at first with 3 pixels, then 5 and finally with 100 pixels size).

(a) (b) (c)

Fig. 22: Mesh segmentation based on the blue edges

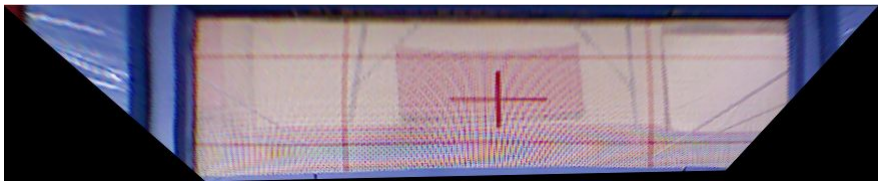
Then we used Canny with a threshold of 0.15 to get the edges and apply the Hough transformation independently for the horizontal and vertical lines.
For vertical edges the specific angle of the lines in the image was from 20 to 60 and -20 to -60. Regarding Hough transform, we specified that the maximum number of peaks to be identified were 20 and the threshold to define the peaks were 30% of the total size of the Hough matrix. Once estimated the vertical lines, we discarded the ones that measure less than 65 pixels and its discontinuity exceeded 10 pixels.
For the horizontal edges was specific that the angles of the lines were close to 0. The threshold for considering the peaks in the matrix Hough remains the same as the vertical edges, however the maximum number of peaks was reduced to 10, while the criterion to discard the lines was increased to 200 pixels long and 40 pixels discontinuity.



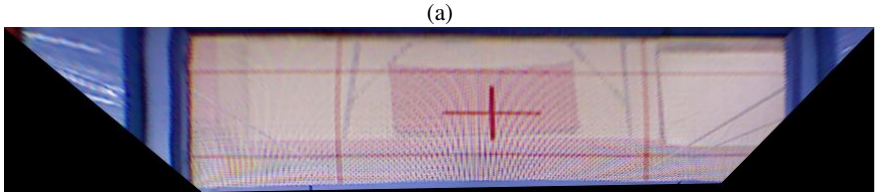
(a) (b) (c)

Fig. 23: Edges of the mesh detected by Hough transformation

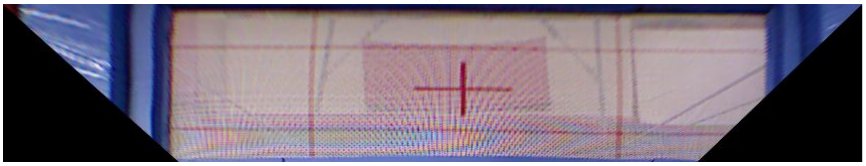
Finally we could estimate the homography and transform the image.



(a)



(b)



(c)

Fig. 24: Images transformed with the estimated homographies

In order to experiment with the landing position estimation, we took a subset of 140 frames. This subset was chosen because it is the piece of sequence in which the person performs jumping movements much closer to an athlete than the rest of the sequence. However some of the jumps are still with the feet very separated and facing the camera, which in an exercise of an athlete is not very usual.

Performing the experiments we had to choose between different parameters and tune them to find the optimal functionality. For the down-sampling method the voxel grid size was 1 cm because with a bigger size too much information was lost. In the euclidean cluster extraction, the clusters we wanted to find were very small and the points quite far between them. Thus the minimum cluster size was 5 and the cluster tolerance 6 cm. For the planar segmentation we only had to be sure that the algorithm segments all the area in the world because there was only the mesh, for that the parameters were as default.

Below is shown the test made, visualizing the 3D mesh and the clusters founded in it with colored in red the ones that belong to the athlete landing. Also is shown the color image corresponding and the 2D mesh with a blue area that indicates where the athlete touches. This let us to compare between the color image where a human can easily determine where the person hit the mesh and the estimation that our system does.

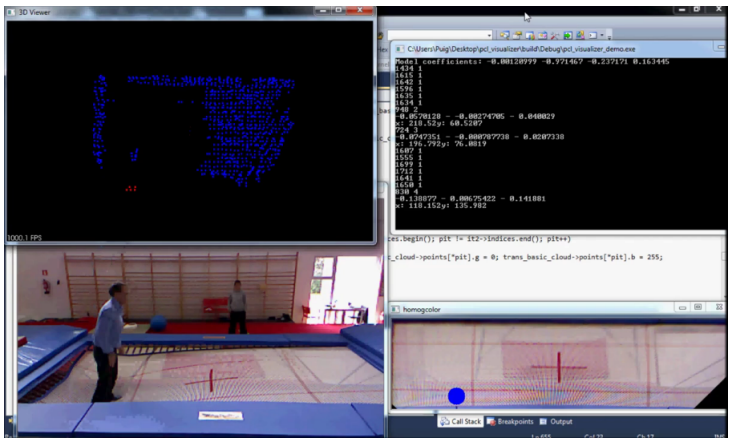


Fig. 25: Experiments on landing position estimation

V. RESULTS

As result of the estimation of the homography in order to undo the perspective in the images, we can see in figure 24 how taking three different images of the trampoline with the camera in the same point, the result achieved is very similar in all the cases. Therefore we have a fairly robust system able to estimate an homography in order to make parallel the lines of the mesh.

Some qualitative results can be seen.

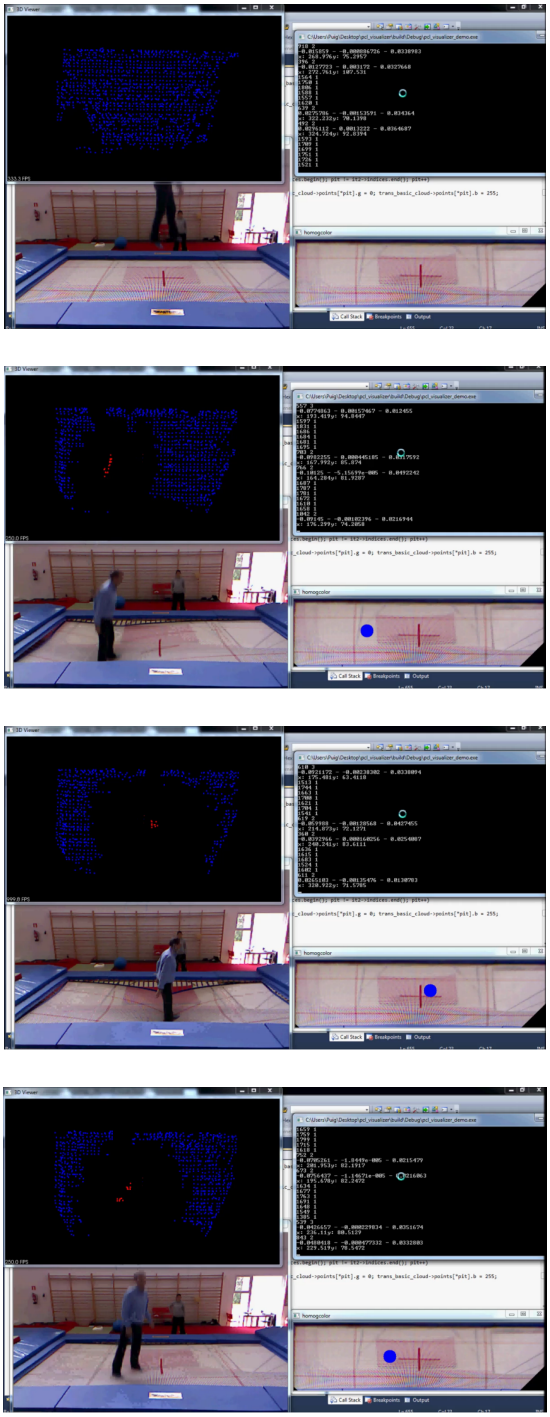


Fig. 26: Images transformed with the estimated homographies

In analysis, the application is robust and is able to estimate with a good precision the landing point of the athlete. As this graphic shows the tendency of the estimation of the 18 jumps in the sequence is normal and good, where normal is a correct estimation within a 20 cm range of the exact real point, good is less than 20 cm an bad more than 20 cm.

Fig. 27: Quality of the location estimation

We also see that the jumps are commonly detected in two frames which imply that two estimations are made of the same jump. This can be merged computing the average position of the frames that concerns the same jump, and in the same way if the frame rate is increased, the number of frames to estimate each jump will also increase, making the estimation more robust to outliers.

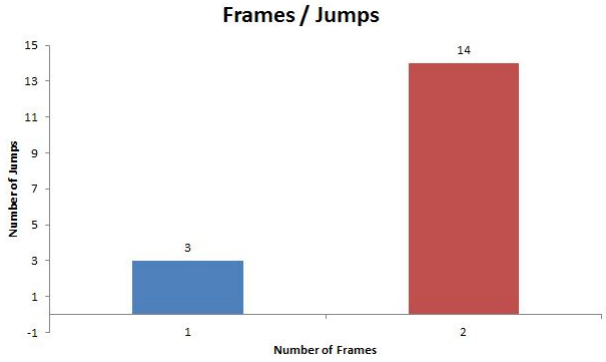


Fig. 28: Number of frames that concern the same jump

VI. Conclusions

In conclusion we were able to get multi-modal RGB-Depth data from a Kinect camera and implement a system that allows the athletes to automatically know their performance in the trampoline sport thanks to the estimation of the landing location. The system is robust and accomplishes the requirements defined by athletes, trainers and judges. Although it is not very precise in the estimation, it was not a requirement because the form of the judges penalizes the athletes for getting out of the painted rectangle is when they almost touch the blue mat.
It is a cheap system that works well and with some arrangements and more tests it can be an extremely useful tool for sports centers.

Regarding the sequence recorded, surely the way that athletes jump and perform the exercises is so far from the jumps that our volunteer perform. As is discussed in the previous sections, this is a inconvenient because we faced some problems that probably in a real exercise would not appear, as the jumps with the feet separated or face to face with the camera, and also probably there are another type of problems that we did not think about, as the height and speed that the athletes reach in the

jumps.

As future work, more field tests are needed to determine which are the details on some requirements. Also is needed to make the system more robust to changes in the scenario, such as the light or camera position, and to errors, defining new strategies and ideas as speed up the frame rate to have more frames in the same jump and make an average of the location. Finally, some specifications could also be added, as the detection of the position of the athlete at the time of the landing or make a history of what types of jumps has done and where he land in each of them in order to compare the different landings of a same exercise, or what has been the trajectory of the contact points.

REFERENCES

[1] Trampolining-Online.co.uk, *Information and facilities for UK Trampolining clubs - Judging for dummies* www.trampolining-online.co.uk/judging/judging-for-dummies/execution-judges.php

[2] Olympic.org Official website of the Olympic Movement, *Trampoline equipment and history* http://www.olympic.org/trampoline-equipment-and-history?tab=history

[3] Gymnastics Australia, *About trampoline sports* http://www.gymnastics.org.au/default.asp?MenuID=Gymsports/20035/0,Trampoline/c20064/3289

[4] Bronson, *Equipamiento de gimnasios - Eurotramp* http://www.bronson.cl/eurotramp.html

[5] Trampolining, *Wikipedia the free encyclopedia* *h*ttp://en.wikipedia.org/wiki/Trampolining

[6] P. F. Luo, Y. J. Chao ,M. A. Sutton, W. H. Peters III.: Accurate measurement fo three-dimensional deformations in deformable and rigid bodies using computer vision. Experimental Mechanics Vol. 33, Issue 2, 123–132 (1993)

[7] Cristina, Federico Dapoto, Sebastian H. Russo, Claudia Cecilia.: 3D tracking and trajectory generation. VII Workshop of Computer Science Researchers, 310–314 (2005)

[8] Wikipedia (June 13, 2014), "Kinect". http://en.wikipedia.org/wiki/Kinect

[9] S. Choppin, S. Clarkson, B. Heller abd B. Lane, J. Wheat, *The accuracy of badminton player tracking using a depth camera* Centre for Sports Engineering Research, Collegiate Hall, Collegiate Campus, Sheffield Hallam University, Sheffield, S10 2BP

[10] S. Choppin, S. Clarkson, B. Heller abd B. Lane, J. Wheat, *The potential of the Microsoft Kinect in sports analysis and biomechanics* Centre for Sports Engineering Research, Sports Technology, 6 (2), 78-85. (2013)

[11] L. Holsti, T. Takala, A. Martikainen, R. Kajastila, P. Hmlinen, *Body-Controlled Trampoline Training Games Based on Computer Vision* CHI '13 Extended Abstracts on Human Factors in Computing Systems, 1143-1148. (2013)

[12] Yan Long Che, Zhong Jin Lu, *The Key Technology Research of Kinect Application in Sport Training* Advanced Materials Research (Volumes 945 - 949), 1890-1893, (2014)

[13] A. Fernandez-Baena, A. Susin, X. Lligadas, *Biomechanical Validation of Upper-Body and Lower-Body Joint Movements of Kinect Motion Capture Data for Rehabilitation Treatments* Intelligent Networking and Collaborative Systems (INCoS), 2012 4th International Conference, 656-661

[14] Chuan-Jun Su, *Personal Rehabilitation Exercise Assistant with Kinect and Dynamic Time Warping* IJIET 2013 Vol.3(4): 448-454 ISSN: 2010-3689 (2013)

[15] Experimedia. http://www.experimedia.eu

[16] Point Cloud Library (PCL). http://www.pointclouds.org

[17] Kinect SDK for Windows. http://www.microsoft.com/en-us/kinectforwindows/