REAL-TIME AUTOMATED CONTOUR BASED MOTION TRACKING USING A SINGLE-CAMERA FOR UPPER LIMB ANGULAR MOTION MEASUREMENT

Alireza Akhavizadegan[1], Mohd Yamani Idna Idris [2*], Zaidi Bin Razak[3], Mazlina Binti Mazlan[4], Norhamizan Binti Hamzah[5], Ainuddin Wahid Abdul Wahab[6], Mehdi Hussain[7]

[1, 2, 3, 6]Department of Computer System and Technology, Faculty of Computer Science and Information Technology, Universityof Malaya, Kuala Lumpur, Malaysia

[4, 5]Department of Rehabilitation Medicine, Faculty of Medicine, University of Malaya, Kuala Lumpur, Malaysia

[7]School of Electrical Engineering and Computer Science, National University of Science and Technology, Islamabad, Pakistan

Email: alireza_a110@hotmail.com[1], yamani@um.edu.my[2*](corresponding author), zaidi@um.edu.my[3], mazlinamazlan@um.edu.my[4], norhamizanhamzah@um.edu.my[5], ainuddin@um.edu.my[6], mehdi141@hotmail.com[7]

DOI: https://doi.org/10.22452/mjcs.vol33no1.4

ABSTRACT

Upper limb angular motion measurement is particularly a subject of interest in the field of rehabilitation. It is commonly measured manually by physiotherapist via goniometer to monitor stroke patient's progress. However, manual measurement has many inherent drawbacks such as the need to have both patient and physiotherapist to be at the same place. In this paper, an automated real-time single-camera upper limb angular motion measuring system is proposed for home-based rehabilitation to aid physiotherapist to monitor patient's progress. The measurement of concern are angle measurement of elbow extension, elbow flexion, wrist flexion and wrist extension. To do this, we propose a method that utilized predefined coordinate points extracted from the contours of the object named as contour based motion tracking. The proposed method overcome problems of prior target tracking techniques such as Kalman filter, Optical flow and Cam-shift. The proposed method includes skin region segmentation and arm modelling for motion tracking. Prior skin segmentation techniques suffer from fixed threshold value set by the user. Therefore, we introduce dynamic threshold based on the lower and upper threshold boundary of isolated skin regions from the background. To ensure the reliability of our skin segmentation method, we compare them with four different related algorithms. The result shows that our skin segmentation method outperformed the prior method with 93% detection accuracy. Following the segmentation, we model the arm motion tracking by formulating mathematical equation of various points and motion velocities to track the arm. We then model the wrist and elbow position to estimate arm angular motion. The method is put together and tested with real human subject and other test settings. The result shows that our proposed method able to produce an accurate and reliable reading of +-1.25 average range of error from actual physiotherapist reading.

Keywords: Angular Measurement, Stroke Rehabilitation, Home-Based Rehabilitation, Upper Limb, Motion Tracking, Skin Segmentation, Contour-based Measurement

1.0 INTRODUCTION

As the public health systems are improving, the demand on the physical disabilities rehabilitation increases. The interest in rehabilitation engineering becomes more common and many rehabilitation instruments are introduced to help patients' recovery. The instruments are designed to ensure rehabilitation exercise are conducted properly and to assess the progress of the rehabilitation. However, most of the current implementations are done manually and normally requires the present of both patient and physiotherapist for the rehabilitation process. Professional physiotherapist needs detailed data to monitor clinical progress and to adjust their treatments or prescriptions based on the patients' progress. On the other hand, the patients need real-time feedback to correct their movements and to motivate themselves. This can be a major hindrance since it involves several problems such as traveling cost, traveling time, difficulty in setting appointment schedule, slow and etc. [1]. Therefore, home-based rehabilitation is considered by prior researchers to

overcome these problems. Home-based rehabilitation has also been seen to be an alternative to oversubscribed international healthcare systems [33].

There are various home-based rehabilitation systems but developments using motion capture systems are gaining more attention. This is due to the fact that motion capture systems have been well received in other field such as gaming, human-computer interaction and surveillance tracking based applications [2-7]. Common motion capture systems are instrumented gloves, wrist-worn laser systems, inertial systems, and optical systems [33]. The instrumented gloves and wrist worn laser system are intrusive and can be difficult for patients with deformities. In contrast, inertial system such as Nintendo Wii and mobile phone with accelerometers, gyroscopes and magnetometers have limited capability to measure movements. This is due to the inertial system only uses the wrist flicking information to estimate other movements.

Optical systems, which consist of camera is considered as good candidate for home-based rehabilitation due to their wide range of view. They can be further divided into marker-based visual tracking system and marker-free visual tracking system. In marker-based visual tracking systems, the identifying process of the region of interest moderately simplifies the actual human motion tracking problem. However, marker-based visual system requires extra configurations and settings such as the placement of the retro reflectors, camera calibration etc. Therefore, it is usually desirable to develop a marker-free visual tracking system instead of the intrusive marker based system. There are also systems consists of single camera configuration, multiple cameras configuration, and optical system combined with high-cost sensors (i.e. acoustic, radar and inertial) [8-12]. Single camera configuration is less expensive compared to the multiple cameras configuration and combined systems.

Taking the advantage of the marker free visual tracking system and single camera configuration, this paper will concentrate on these two elements in the design. In this paper, we propose an upper limb measuring system for home-based rehabilitation purpose through visual tracking method. It goes beyond typical gesture recognition that is used only to track the movement of a subject. The proposed system, on the other hand, is intended to attain specific measurement for home-based rehabilitation application. Four angular motion movements which include elbow extension, elbow flexion, wrist flexion and wrist extension, a measurement that is normally done manually by the physiotherapist via goniometer is considered for the system. The angular motion measurement is important attributes to assess the progress of stroke patients. One of the method to measure these four attributes is by utilizing target tracking algorithms such as Kalman filter, Open flow and Cam-shift. However, they seems not to be suitable for our upper limb measurement application since they are unable to track the targeted points. This is due to high computational time is required to keep up with the targeted points and the interferences from the background scene affects the overall performance of the technique. For this reason, we design our measuring system based on the coordinate points extracted from the contours of the object named as contour based motion tracking. Our algorithm overcome prior target tracking techniques by only tracking predefined points extracted from the object contour. The proposed contour based motion tracking is divided into skin segmentation and arm modelling for motion tracking. The skin segmentation separates the upper limb (i.e. the arm) from the unwanted background of the image. Conversely, the arm modelling is used to model the points needed for measuring the four attributes.

In the skin segmentation stage, we consider methods that could detect effectively the arm region. We utilize face recognition technique which act as an extra clue to get more accurate results on skin segmentation. One of the major problem in prior skin segmentation technique is the use of a fixed threshold value set by the researchers. The fixed threshold will not able to cope with different skin colors and different illumination conditions which lead to inaccurate segmentation result. There are also adaptive thresholding method that separates skin from non-skin pixels. They are based on a color histogram that able to cope with the rapid illumination changes. However, the drawback of this method is that the threshold values are bound to change if different human races and wider range of varying illuminations are to be taken into account. For this reason, we propose a method that adjust the best values for the lower and upper threshold boundary of isolated skin regions from the background to overcome the problem. The method has two significant advantages where the process of initializing the threshold values becomes automated and it allows the algorithm to dynamically find the best threshold values based on individual skin color. We also utilize the hue, saturation, and value (HSV) space instead of red, green and blue (RGB) color space. The HSV provides a high separation contrast between skin regions and non-skin regions. Our proposed method is tested and compared with four other related skin segmentation methods such as skin probability map, HSV back-projection, YCrCb back-projection and RGB back-projection algorithm.

In the arm modelling for motion tracking, we formulate mathematical equation to model the points needed for the upper limb measurements. Based on the predictability of spacing of the points of the arm's contour, we are able to estimate any point on the arm by assigning the appropriate coordinate point. The proposed method is then tested with human subject and compared with conventional goniometer measurement, which is manually measured by professional physiotherapist. We also include several test at different viewing angles with respect to goniometer to ensure the reliability of the proposed method.

The rest of the paper is organized as follows. The existing human visual motion tracking methods with various configurations are discussed in Section 2. The proposed system is presented in Section 3. Section 4 contains the experimental results and discussions. Conclusions are drawn in Section 5.

2.0 REVIEW OF LIMB MOTION TRACKING SYSTEMS

In this section, we discuss various human motion tracking systems that aimed for medical rehabilitation. L. Enrique et al. [13] developed a prototype called "Gesture therapy" based on two webcams. First, a webcam was placed in front of the patient and the other webcam was placed above the patient's head. Further, patients were asked to interact with a virtual environment by moving their impaired arm accordingly. The system tracked the hand using color and motion information in 3D space and the coordinates of the hand were sent to a simulator to measure the angles and daily life activities. The multiple camera-based visual motion tracking systems are considered as high-cost and required specific arrangements of cameras instead of single camera-based systems. Conversely, a single camera-based visual tracking systems are low-cost and has advantages of simplicity. However, it suffers from various constraints, i.e. prior knowledge about the appearance of the subject, the geometry, kinematics, and dynamics of the subject.

In a single camera-based configuration, Polana et al. [14] proposed an approach for visual motion tracking which can spatially and temporally normalize, segment and recognize a repetitive motion i.e. walking or any repetitive motion activity without having any prior knowledge of specific parts. Following the approach, Ming Du et al. [15] proposed a system based on monocular camera to track the human motion by taking the advantages of the DE-ME (Markov chain) algorithm. In addition, this algorithm system solved the 3D articulated model-based human motion tracking problem. Similarly, in [16], Ding et al. proposed a method that estimates the relative 3D coordinates of skeleton joints with two major limitations. First, in the proposed model, the targeted depth value was smaller than the distance between the subject and the camera. Second, the human movement must be faced to the camera. Furthermore, Liu et al. [17] presented a full-body human motion tracking system using exemplar-based conditional particle filter. Similarly, Spruyt et al. [18], presented an unsupervised method to automatically learn the context in which a hand was placed. Adams et al. [19] used a Kinect sensor and a high-fidelity virtual world interface to acquire the depth image of the scene. Further, it applied an unscented Kalman filter-based tracking algorithm to estimate upper extremity joint kinematics in real-time during performance of virtual activities of daily living. The proposed system was able to generate metrics related to speed and smoothness of motion for subject, however, this system requires special hardware.

Tian et al. [20] addresses the limitations of Kinect and Inertial Measurement Unit (IMU) sensors used in trajectory tracking systems and proposed a method that fuses IMU and Kinect data to provide robust hand position information. Where the proposed system achieved hand position by three fusion strategies: double integration of IMU internal sensors, IMU internal sensor fusion with geometrical constraints and unscented Kalman filter (UKF) based fusion of IMU and Kinect. Similarly, Tanaka et al. [21] proposed a portable six degree of freedom motion tracking system comprising of a high-accuracy augmented reality markers that would be attached to the subject arms and a single camera. This system was able to estimate the pose of the arms with an error of 5_{mm} and 2^0 (degree) in space and orientation, respectively. Ligorio and Sabatini [22] developed a novel Kalman filter for human motion tracking by fusing the data from a triaxle gyroscope and a triaxle accelerometer. The approach used stereo photogrammetric data as a reference, and the average root mean square attitude error of 3.6° in manual activities and 1.8° in local motion tasks.

From our study most of the tracking systems mentioned earlier employ special hardware which increases the overall cost of system, and has less portable capability. Despite the different types of sensors used in tracking the motion as mentioned above the core common algorithms that are used to perform the motion tracking are: Kalman filter, Optical flow, and Cam-shift. In next few paragraphs, we will briefly discuss the basics of these techniques.

A. Kalman Filter

Kalman filter is an efficient recursive filter that roughly calculates the state of a dynamic system from a series of incomplete and noisy measurements [22]. The kalman filter has two unique stages: predict and update. In the first stage it uses previous time-step to generate an estimate for the current state. In the second stage, the information from the current time-step is used to refine the prediction to compute a more accurate estimation. The accuracy of the prediction in the kalman filter depends on how accurate the previous information was computed. However, in a vision-based arm motion tracking system, where the frame rate of the webcam is about 30 frames per second, so, this sampling rate may not be sufficient to estimate some arm maneuvers. As our experiments on Kalman filter motion tracking shows this fact in Fig. 1. When we applied Kalman filter tracking algorithm to track a desired point on the tip of the subject's finger as shown in Fig 1(a), next as the subject moves his arm back and forth the targeted point, it failed to keep the subject's movements. Furthermore, it takes more times due to computation and performed slowly upon any displacement of the targeted point.

Fig 1: Kalman filter tracking. (a) Initial position; (b) Failure in tracking.

B. Optical Flow

Optical flow motion tracking algorithm is considered as another popular object tracking technique used in computer vision. The algorithm is based on the fact that lighting condition in two successive frames is relatively constant, thus the intensity of each pixel on the object would be the same [25]. Based on this assumption if the position of specific pixel intensity changes in the next frame we can conclude that the point on the object has moved in the time in between two frames. As shown in Fig 2, we applied an optical flow tracking algorithm for the purpose of motion tracking to see how it copes with the motion of the arm. In our experiments, we found that this method fails to successfully track the targeted point. Due to the nature of its algorithm the assigned targeted point drifts away over time or by any rapid movements of the arm thus, resulting in an unreliable reading in Fig 2 (b). Moreover, assigning a precise location to the optical flow algorithm to track can be challenging because it tends to drift the assigned point slightly away from the intended region which can be an issue in some situations.

Fig 2. Optical Flow tracking (a) Initial position; (b) Failure in tracking.

C. Cam-Shift

Cam-shift algorithm is an adaptation of mean shift algorithm that basically differs from the mean shift algorithm. The search window adjusts itself in size based on how far or closer the object moves from the camera. The concept behind cam-shift tracking method is simple. Where it starts with a set of distributed pixels in an image and a small rectangular window, the objective is to move the window to an area in which the pixel density is at its maximum (basically where there are maximum numbers of pixels) [30]. The drawback of this tracking technique is that it won't work when the desired tracked object is close to black or white color and this is due to its method of tracking which uses the hue component of the image. Additionally, any rapid or sudden changes to the motion of the object can make the system to fail.

Fig 3: Cam-shift tracking (a) Initial position; (b) Failure in tracking.

As shown in Fig 3, we implemented Cam-shift algorithm to see how it can cope with our arm motion movements. First of all, when it comes to motion tracking with Cam-shift algorithm we are not able to assign a single point to the algorithm and ask it to track just this specific point because simply it doesn't accept a single point. Hence, we have to assign a region of interest to the algorithm as illustrated in Fig 3(a), as the subject moves his arm the elliptical shape representing the area that the algorithm is tracking drifts to another area. As it is demonstrated due to the fact that it drifts a way to another region it was considered not suitable for our purpose system.

3.0 PROPOSED UPPER LIMB MOTION TRACKING SYSTEM

In this section, we introduce a single camera based marker-free upper limb measuring system for the purpose of home-based rehabilitation. The proposed method employs the state of the art face recognition algorithm to identify the face as a clue of skin isolation known as automated skin segmentation process. After segmenting the skin pixels it further introduced contour based motion tracking of arm region. Finally, proposed system estimate the wrist and elbow joints positions based on contour information, where these positions used as reference point for evaluating the angular motion performed by the arm i.e. wrist flexion, wrist extension, elbow flexion, and elbow extension. The detail of the proposed system with different stages is shown in Fig 4 and discussed in section 3.1 to 3.4.

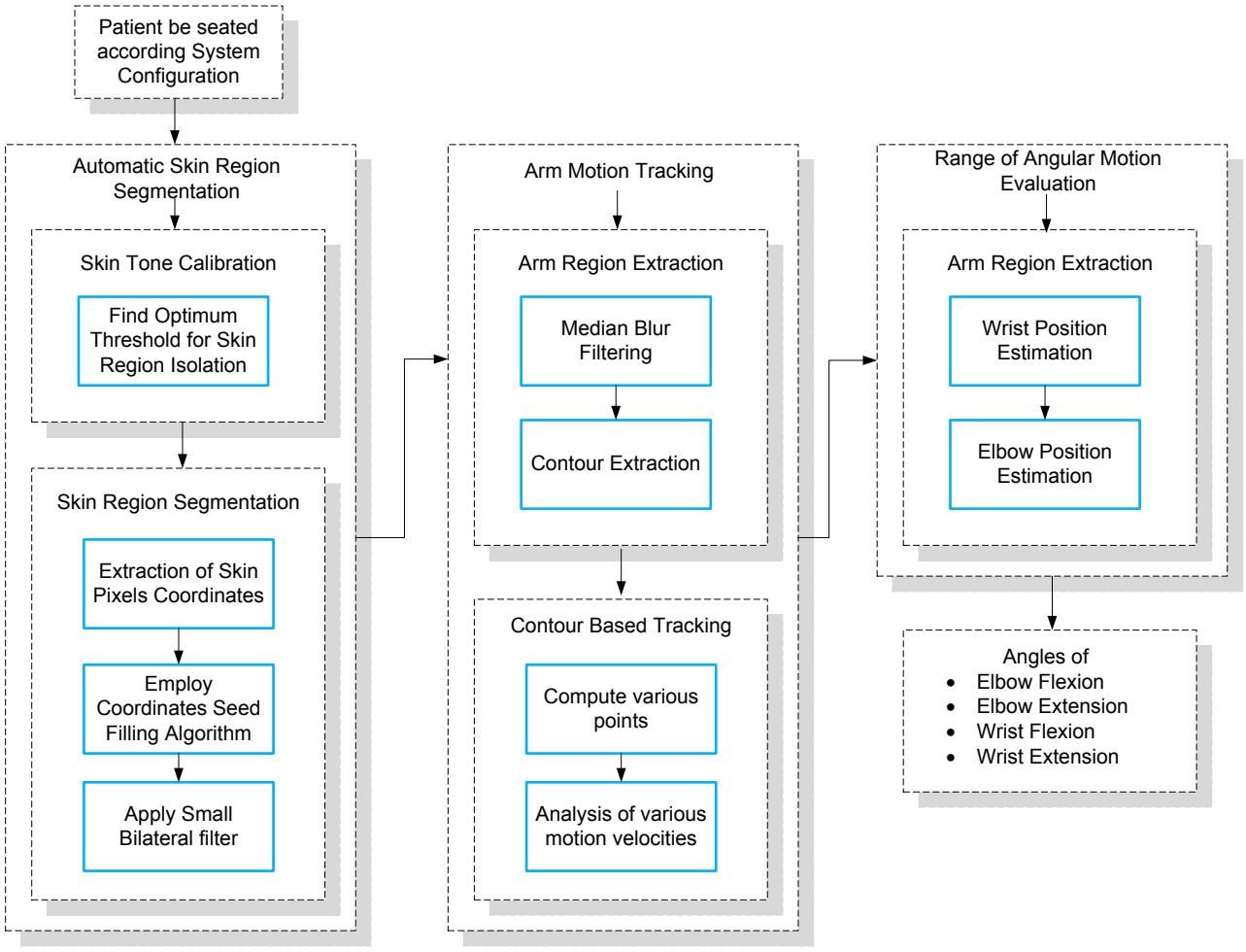


Fig 4: Flowchart of Proposed Upper Limb Motion Tracking System

3.1 System and Use Case Configuration

One of the motivations of the proposed system is a low-cost motion tracking solution without any extra hardware, where the proposed system employ the built-in webcam of a laptop for image capturing and processing for various tasks. In this system configuration setup, first, the subject has to be seated on a chair approximately one meter away from the webcam as shown in Fig 5(a). In this stage, the proposed system apply the skin tone calibration procedure, in which only the subject's face should be targeted in front of the webcam. Once the proposed system identified the subject's face with his/her skin tone calibration, in next stage, the subject is required to be moved either to the left or right of the camera field of view and show his/her targeted arm in front of the webcam. Fig 5(b) shows an example of a subject's arm, where he/she can flex various position of an arm according to physiotherapist recommendation. Then, the proposed system

apply its internal contour based motion tracking processes that indirectly computes the parameters such as the rate at which a particular point on the arm has moved from one point to another (the velocity of a point) and the range of motion (ROM) performed by the arm, namely wrist flexion, wrist extension, elbow flexion and elbow extension. The rest of proposed system stages are briefly discussed in the next section.

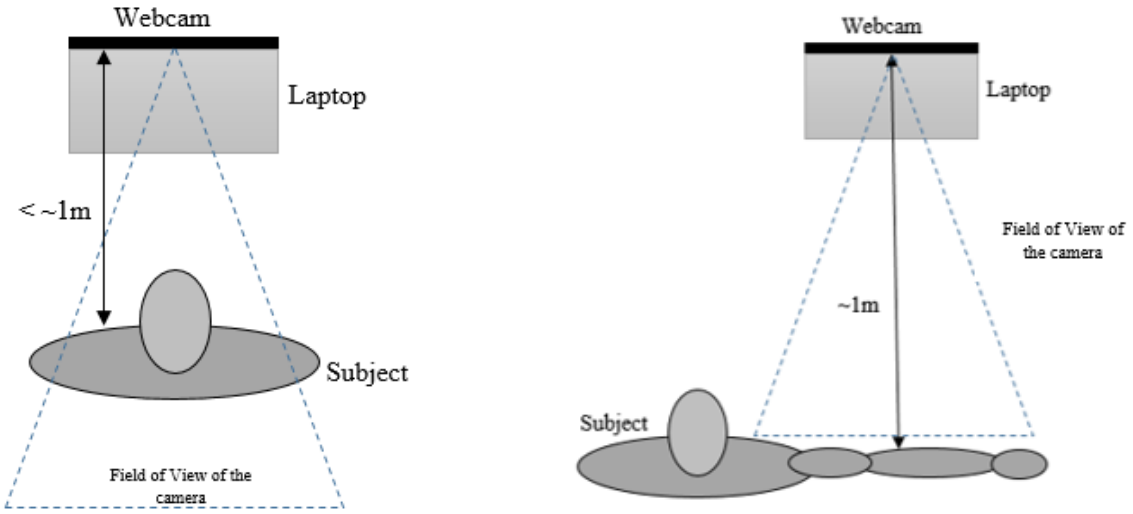


Fig 5: (a) Top view of the calibrations procedure, (b) Top view of the experimental setup

3.2 Automatic Skin Region Segmentation

Skin detection has an important role in a wide range of image processing applications such as gesture recognition, human tracking, content based retrieval systems, and different human computer interaction domains [12]. Skin detection methodologies based on skin-color information is a powerful and often computationally inexpensive cue. Furthermore, it is robust towards geometric variations (rotations), scaling and partial occlusion. In general, however it is not an easy task to extract regions of specific color from a known color image, since the appearance of skin color information varies upon several factors such as, the camera sensor characteristics, ambient illumination, ethnicity, individual characteristics (such as age, makeup, hairstyle glasses and etc.). Thus, due to these variables the outcome of the skin segmentation will be affected and consequently produces high false positive rate. Therefore, an algorithm which copes with these variations should be developed.

Many different skin modeling and classification approaches have been proposed to detect skin-color in images, however, most of the reported works showed good results only for a limited set of illumination conditions and skin types. The adaptability of a skin-color detection algorithm to the changes in the ambient lighting and the viewing environment is one of the important factors in determining its success. There are mainly two different classes of approaches that deals with the rapid changes in ambient lighting conditions for skin detection: 1) color constancy and dynamic adaptation. Various skin-color constancy approaches such as [4, 2, 5, 3, 6] have been proposed, however all these methods require the presumption of the camera characteristics, the illuminant properties or the distribution of the color values. On the other hand, dynamic adaptation approaches seem more promising and reliable in dealing with rapid illumination changes [12]. For example, in dynamic adaptation domain, Cho et al. [1] proposed an adaptive thresholding method in Hue, Saturation and value (HSV) color space that separates skin from non-skin pixels using a thresholding box, the threshold values of the S and V components are updated based on a color histogram built in SV color space to cope with the rapid illumination changes. However, the drawback of their method is that the threshold values are bound to change if different human races and wider range of varying illuminations are to be taken into account.

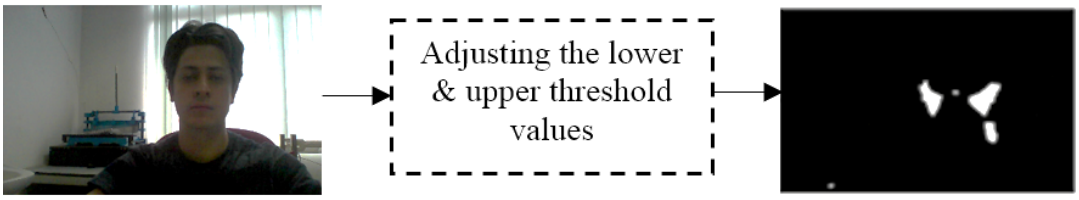
The purpose of this stage is to introduce automatic and independent initializations setups for subject/patient instead of any manual hassle for setting up the system. For this, we first employed the skin tone calibration and further applied the segmentation process to identify the subject's face skin region. This skin region based information used as input of contour motion tracking process. Generally, in literature, various skin segmentation techniques proposed by prior researchers such as skin-color constancy methods and adaptive thresholding methods. These methods shows acceptable results only for a limited set of illumination conditions and skin types. However, we proposed a process to find the best adaptive lower and upper thresholds, which isolate the skin region from background. This process employed the hue, saturation, and value (HSV) space instead of red, green and blue (RGB) color space.

3.2.1 Skin Tone Calibration

The skin tone calibration procedure runs only once at the start of the program. It utilizes the rapid face detection method introduced by Viola and Jones [26] to robustly isolate the face region within the scene, next step we automatically adjust the lower and upper threshold value of our HSV color space until we reach an acceptable number of skin pixels present with in the detected face region. By applying this technique, we are able to eliminate the non-skin regions of the scene from the skin regions

3.2.1.1 Finding the optimum threshold values that best isolates the skin regions from the background

At this stage, the aim is to identify the best values for the lower and upper threshold boundary of isolated skin regions from the background. The 'best lower and upper threshold value', means the threshold values that are after being applied to each frame which preserves the maximum number of skin pixels presented in the scene. Fig 6(b) shows the outcome of this stage in which all the background (non-skin pixels) are eliminated, while portions of the skin region that corresponds to skin pixels are retained.



(a) (b)

Fig 6: (a) Input image (b) Binary representation of the skin regions (as white pixels) extracted from our repetitive sequential thresholding procedure.

To identify the lower and upper threshold values that can best isolate the skin regions from the background two different aspects of each frame are taken into account. (1) The number of skin pixels within the scene as shown in Fig 7(a). (2) The number of non-skin pixels within the scene as shown in Fig 7(b).

(a) (b)

Fig 7: (a) Representation of skin pixels (b) representation of non-skin pixels.

We refer to the extracted portion of the person's face to be the skin region since we are sure that this region contains skin pixels of the person. Furthermore, we refer to the entire scene excluding the extracted face region as the non-skin regions as shown in Fig 7. Afterwards, it automatically adjusts the lower and upper threshold values sequentially (one by one, starting from 0 up to 255 for the lower threshold boundary, and from 255 down to 0 for the upper threshold) and applies it to the entire image. Subsequently, we observed the outcome that shows the effect of the thresholding on each frame by taking into account the changes in the total number of skin and non-skin pixels within the scene. Next, we repeat the process of adjusting threshold values and observing the outcome until we achieve a fair amount of separation between the skin and non-skin pixels within the scene. At this point, we apply a morphological operation filter to eliminate any unwanted small regions presented in the non-skin pixel image representation that is incorrectly identified as to be skin region due to the similarity of the particular region to the skin. The end result is a binary image that shows the skin pixels identified based on lower and upper threshold values as shown in Fig 6(b). From this point onwards the threshold values are obtained, that will be used recursively throughout our skin detection algorithm. Initializing the threshold values for segmenting the skin regions by face detection has two major advantages:

(1) The process of initializing the threshold values becomes automated.
(2) Allows the algorithm to dynamically find the best threshold values based on individual skin color.

3.2.2 Skin Region Segmentation

The whole segmentation procedure carried out using two sub-stages.

- Extractions of skin pixels coordinates
- Feeding of extracted points to seed filling algorithm

First, it extracts the coordinate points of the skin pixels from skin tone calibrations stage as shown in Fig 6(b). Second, it feeds the extracted skin pixel location to a seed filling algorithm. After this stage, the seed filling algorithm locates each point within the scene and uses it as a reference (seed) point to isolate an area in a multi-dimensional array based on the color similarity of the neighboring pixels. The seed filling algorithm looks at the pixels associated with each of the coordinate points (known as the seed point) and takes its hue information as a reference then the hues of each of its eight neighboring pixels are compared to the reference point. If the difference is within the acceptable predefined range, it is marked as skin otherwise this considered a non-skin pixel. In practically the hue, saturation and brightness are likely to be different in various parts of the skin regions such as the face and/or the arms. Hence to overcome this issue, we take advantage of bilateral filter. Bilateral filter is known to be an edge-preserving smoothing yet time-consuming filter. Thus, to take advantage of its edge preserving and smoothing features in real time we tackle in two parts:

- First, reduce the image resolution.
- Second, apply several small bilateral filters.

In the first part, by reducing the image resolution basically means lesser number of pixels used and save the time for processing. Further, additionally applying several small bilateral filters rather than one large filter will allow the filter to run in real time.

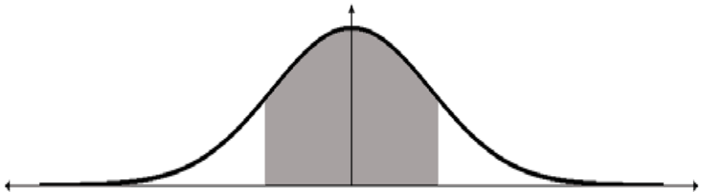
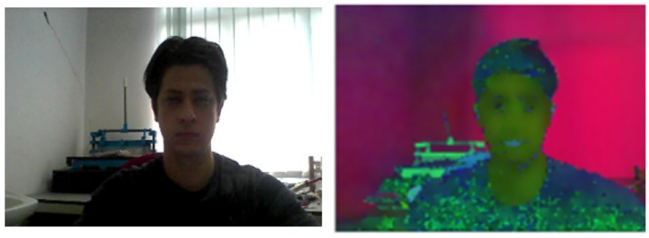


Fig 8: Illustration of bilateral filter truncating.

The concept of truncating the filter is illustrated in Fig 8. If we assume the filter matrix (kernel) of our bilateral filter to be 21 x 21 pixel wide shown here as the bell curve, consequently, if we use the minimum kernel size of 9 x 9 pixel wide (shown above in gray) we still are able to cover the major parts of the filter without wasting time on the minor parts of the filter (the white area under the curve), consequently the filter will run several times faster. The result of the proposed accelerated bilateral filter is shown in Fig 9. As it is depicted from Fig. 6(b) the face region is highly distinguishable from rest of the scene, while the hue variation within the face region is more persistence and predictable.



(a) (b)

Fig 9: (a) Input image (b) bilateral filter applied to the input image.

However, the complete procedure of our automatic skin region segmentation is illustrated in Fig 10. In Fig 10 (a), it received the input images and this stage only runs once at the start of the program. Then the lower and upper threshold values are sequentially adjusted and applied to the image recursively until finding the optimum value that separates the best skin region from the non-skin regions as shown in Fig 10 (b). Finally, we applied a seed filling algorithm to the image as shown in Fig. 10 (c) using the skin location information from the previous stage, so it isolates the skin regions within the scene as shown in Fig 10 (d).

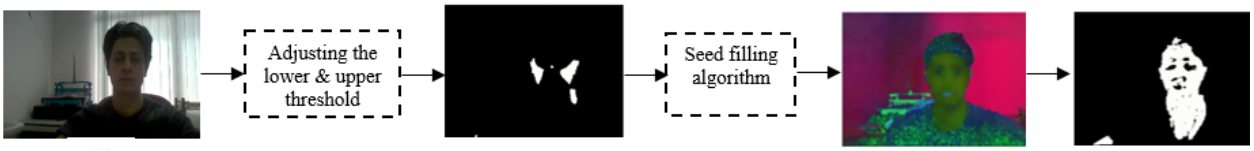


Fig 10: The segmentation procedure; (a) input image; (b) thresholded skin; (c) seed filling output; (d) skin region extraction.

3.2.3 Skin Region Segmentation benchmarking

In order to evaluate our segmentation algorithm's performance, we used the publicly available MUCT dataset as our ground truth, the subjects in the dataset vary in age and ethnicity and are captured under a number of different lighting conditions and head poses. Masks have been segmented manually for over 300 carefully selected photographs from the dataset, the photographs where cautiously selected in order to ensure all possible races with different ages, genders, different types of clothing, hairstyles and different appearances (such as makes ups, glasses, earrings, etc.) are taken into account. All presented results were obtained from the same set of images. We have measured three parameters for each output of the four different algorithms including our own algorithm based on the ground-truth data from the dataset separately:

- True Positive rate (TPR), is the percentage of correctly detected skin pixels to the actual number of skin pixels in the image.
- True Negative rate (TNR), is the percentage of correctly detected non-skin pixels to the actual number of non-skin pixels in the image.
- Accuracy (ACC), is a number that represents the degree of closeness of the overall output of the algorithm to the actual true output. For example, a value of 1, means that the overall skin and non-skin pixels detected by the algorithm (with respect to the actual skin and non-skin pixels) is completely accurate.

Table 1 True positive, true negative, and accuracy values for various skin detectors

Skin detector Methods	TPR (%)	TNR (%)	ACC
Soriano and Martinkauppi method [28,32]	61.22	94.54	0.879
Soriano and Martinkauppi HSV modified version	78.84	95.56	0.922
Soriano and Martinkauppi YCrCb modified version	64.57	92.45	0.869
Skin Probability Map [27,31]	32.97	94.71	0.827
Proposed Method	85.2	95.2	0.932

Based on our results only 18 out of 300 images were unsuccessful to segment the skin region correctly. This was due to two factors, one, the boundaries between the skin and the background was not successfully isolated hence after applying the seed filling algorithm the segmentation procedure extended beyond the skin to the background, and secondly due to lighting and skin color variation, a huge amount of deviation was implied on the skin region. Hence, causing big differences to appear on the skin region, therefore the seed fill algorithm was not able to cope with the amount of variations thus failed. But beside these, as it is obvious from the results in table 1 our method is able to detect skin regions regardless of the person ethnicity, individual appearance (such as beard, glasses hairstyle etc.), and illumination conditions. Fig 11 shows some examples which the algorithm failed to segment the skin regions.

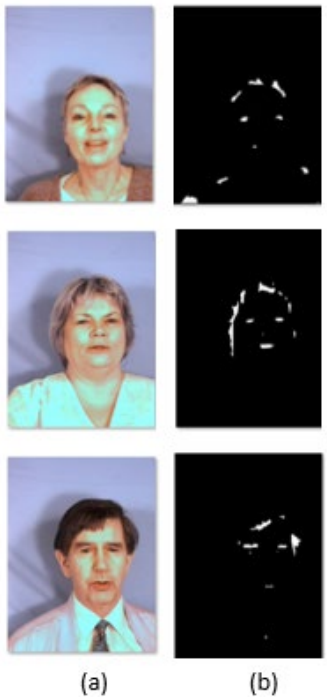


Fig 11 examples of the images that our algorithm failed to segment. Column (a) input images, column (b) failed results

We tested our method with other skin detection techniques such as, Skin probability map [27, 31], Soriano and Martinkauppi [28, 32] adaptive skin detection method. Additionally, we derived and implemented two other algorithms for skin detection based on Soriano and Martinkauppi adaptive skin detection technique. One modified method implements Soriano and Martinkauppi technique in YCrCb color space and the second developed method implements the Soriano and Martinkauppi technique in HSV color space. Fig 12 shows some results of our method and the conventional ones.

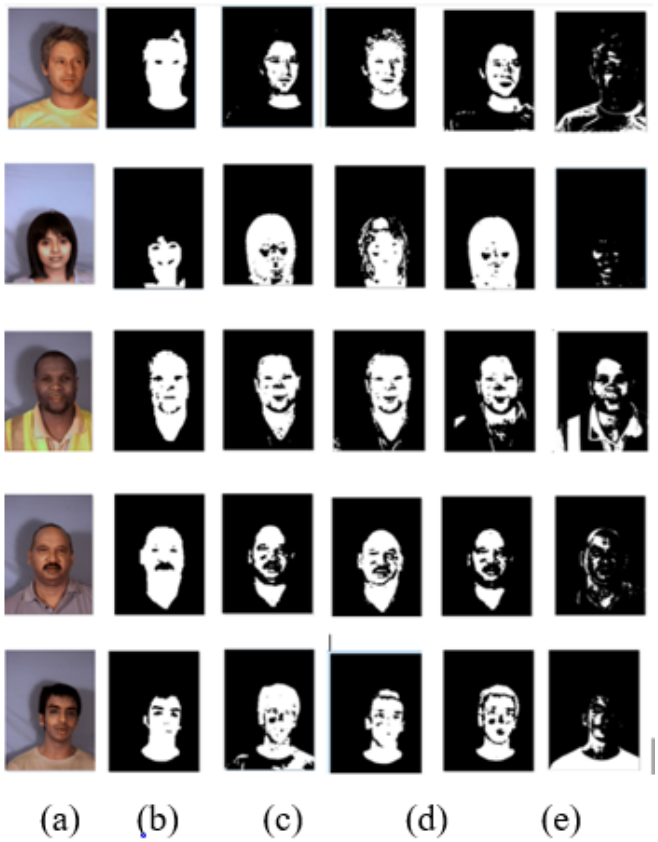


Fig 12 Examples of the skin color detection results: column (a) input images; column (b) our method; column (c) Soriano and Martinkauppi [28,32]; column (d) HSV back projection; column (e) YCrCb back projection; column (f) Skin probability map [27,31].

Since the threshold of the skin probability map (column (f)) method is fixed, the algorithm is not able to cope with different skin colors and different illumination conditions, and overall shows an inaccurate results compared to the rest of the methods conducted in our experiment. Among the back projection methods used and also by referring to table 1, HSV color space back projection showed a more promising results, with 78.84% correctly identifying actual skin pixels, but despite its high TPR, our method outperforms the HSV back projection technique, with a TPR of 85.2%, More importantly, the results show that the algorithm described is much more robust than the other algorithms even if used in different lighting conditions and is able to cope with any skin color variations. Fig 13 illustrates the effectiveness of our skin segmentation compared to four other skin segmentation algorithm mentioned earlier.

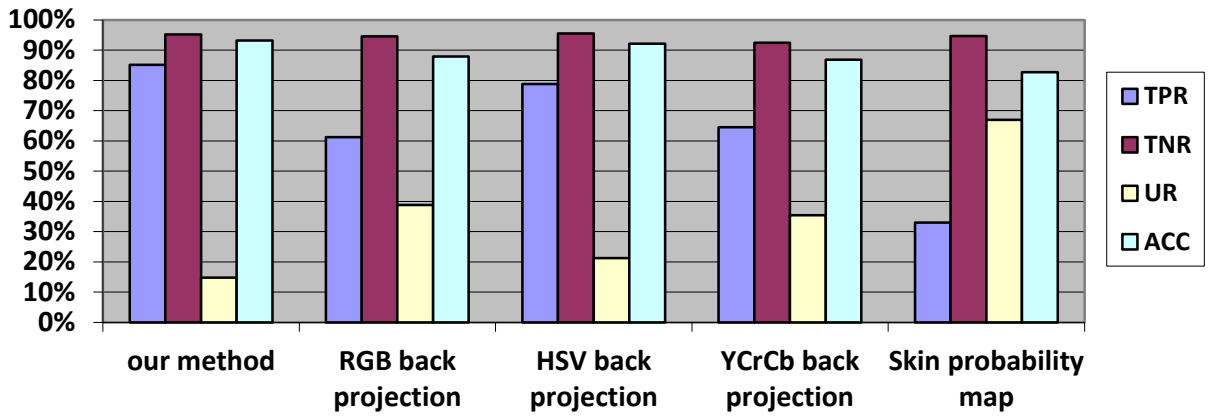


Fig 13 Comparison of skin segmentation effectiveness

3.3 Arm Motion Tracking

The proposed arm motion tracking algorithm is dependent on the correct and accurate segmentation of the skin region as derived from the previous stage. The proposed motion tracking algorithm employed the coordinate information of the arms contour (which would be extracted in this stage) in each successive frame to keep track of a specific point on the arm. The basic constraint of this stage is that we assumed that the subject only shows his/her arm in front of the camera according to illustrated in Fig 5(b). The proposed arm motion tracking consists of two components.

- Arm region extraction
- Contour-based tracking

3.3.1 Arm Region Extraction

This part of the algorithm aims to isolate the arm region completely from the background to avoid any errors in the motion tracking phase. At this stage, we employed the median blur filtering on the image to reduce noise and small unwanted regions for an enhanced representation of the arm. Next, we applied a contour extraction process for two purposes, firstly to filter out unwanted regions within the scene and secondly, to denote the boundaries of the arm for further processing.

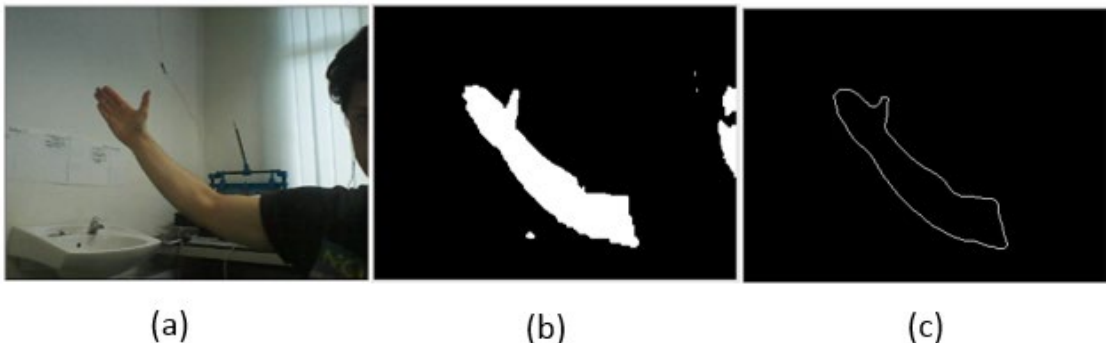


Fig 14: Arm region extraction.

As shown in Fig 14(a), the subject presents his arm in front of the camera. Next, we threshold the entire scene based on the lower and upper threshold values obtained from the previous stage. Further, we achieve a binary representation

image of the skin regions as shown in Fig 114b). At this stage, we can eliminate unwanted regions (example a white spot below the arm and small portion of the face as shown in Fig 14b)) and preserve the desired regions (in this case the arm) by applying a contour extraction algorithm on to the binary image (Fig 14(c)). Further, filtered the regions based on the size of their contour. The size of evaluation is based on the number of points that defines the particular object. Since we assume that the person's arm is situated near the camera (approx. 1m), hence the arm would have the biggest contour instead of other objects as compared to the arm contour. The final result of the arm region extraction stage is shown in Fig 14(c).

3.3.2 Contour Based Tracking

After successfully extracting the contour of the arm as shown in Fig 14(c), the final step is identifying the particular point of interest on the arm for the purpose of tracking. Further, these coordinate points are recursively passed to the next successive frame. The contour of the arm is a sequence of points, which represents the boundaries of the arm. If vector **C** is the extracted contour of the arm it can be expressed as

$$C = \{P_1, P_2, P_3 \ldots\} \tag{1}$$

Where $P_1 = (x_1, y_1), P_2 = (x_2, y_2), P_3 = (x_3, y_3)$ and so on are the coordinate points corresponding to a specific point on the arm. In this paper, we refer coordinate points as a specific column and row on the 2D. For example, a coordinate point of (100, 30) refers to the point on the image, where the 100th pixel (starting from the top left corner of the image as the origin of our coordinate system) in the horizontal line (x-axis or column) meets the 30th pixel from the vertical line (y-axis or row) of the image. Similarly, if we tend to track the tip of the fingers, we assign the algorithm to only pass the coordinate point associated with the finger tips on the arm contour (in this case P_1 which is located at the tip of the fingers) to the next frame each time. The points on the arm contour starts from the tip of the finger P_1(shown in Fig 15) and sequentially progresses clock wise until it reaches P_1 again. Due to this predictability of spacing of the points of the arm's contour, we are able to estimate any point on the arm by assigning the appropriate coordinate point. Fig 15 shows how we have extracted four points on the arm contour namely the finger tips, forearm_1, biceps and forearm_2 as our reference points for further calculation. Below shows the mathematical expression on how we calculate the points as mentioned.

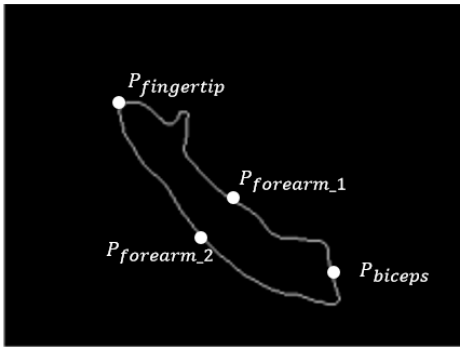


Fig 15: Samples of points estimated on the arm

Where $P_{fingertip}$ is the first point on the arm contour, $P_{forearm\_1}, P_{biceps}, P_{forearm\_2}$, are expressed as follow.

$$P_{finger\ tip} = P_1 = (x_1, y_1) \tag{2}$$

$$P_{forearm_1} = \frac{Total\ number\ of\ contours\ point}{4} \tag{3}$$

$$P_{biceps} = \frac{Total\ number\ of\ contours\ point}{2} \tag{4}$$

$$P_{forearm\_2} = \frac{Total\ number\ of\ contours\ point}{4/3} \tag{5}$$

The velocity of any point on the arm can be obtained by taking into account the duration it takes for the point to move from one location to another in each consecutive frame, which can be described as

$$V = \frac{\Delta d}{\Delta t} \tag{6}$$

Where **V** is the velocity of the motion, $\Delta \boldsymbol{d}$ is the displacement or the amount of shift the targeted point makes in space at each successive frame, and Δt is the duration it takes for each frame to be executed multiplied by the number of frames (nF) at the instance when the motion ends. In our case with the Lenovo ideapad S410p Intel core i7 CPU built in webcam each successive frame is executed every 0.0016s, and the displacement and duration of a motion is calculated as

$$\Delta d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{7}$$

$$\Delta t = (0.0016 \times nF) \tag{8}$$

Where $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ is the distance between the start and the end points of the completed motion. Fig 16 illustrates a simple superimposed wave motion of the arm starting from P_1 and ending at P_2.

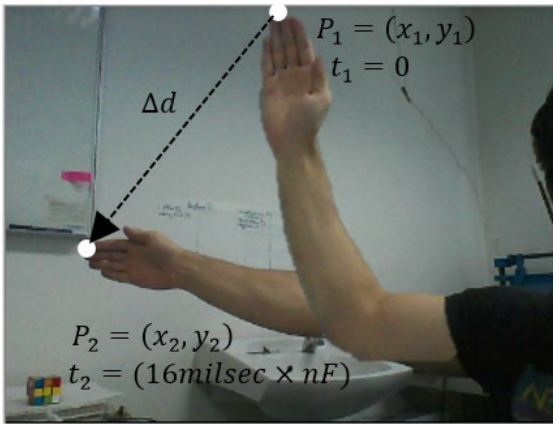
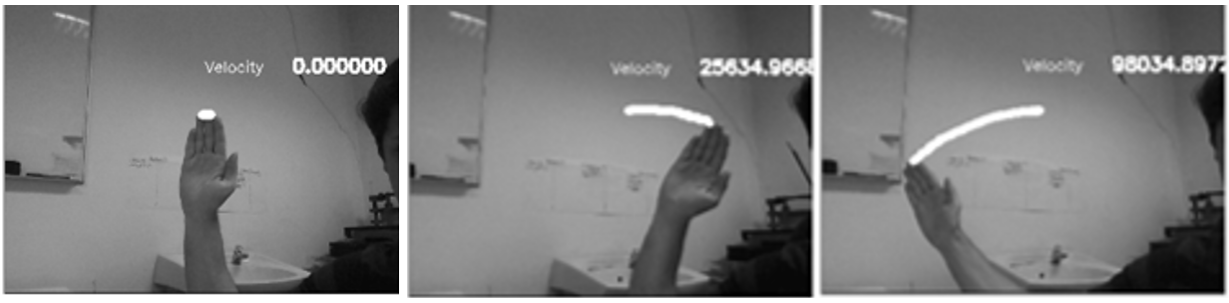


Fig 16: Velocity evaluation.

The calculated velocity has a unit of pixels per second and its numerical value represents the rate at which a point has moved in space with respect to frame reference. In order to have a perspective of how fast or how slow the velocity of the motion actually is, we can either scale down the corresponding value by dividing the whole number by an integer or use a calibrated camera in order to have physical units of measurements. Fig 17 shows three different types of motion, we can identify using our velocity calculations. They are no motion, a slow-motion act, and a fast motion act.



(a) (b) (c)

Fig 17: Velocity evaluation. (a) No motion; (b) slow motion; (c) fast motion.

3.4 Range of Angular Motion Evaluation

To evaluate the range of angular motions performed by the arm, we construct a skeleton structure comprising of two segments linked by a revolute joint (see Fig 18) as a representative of the arm model. No shape modeling is required in our method (example modeling the limb as truncated cones). Two degrees of freedom (DOF) are assigned to each segment.

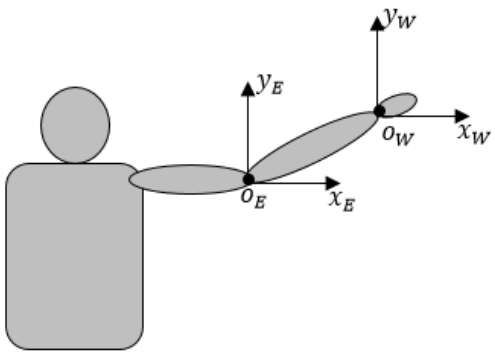


Fig 18: Arm coordinate system

Where x_E and y_E are the two axis used for the angular motion calculation of the elbow, and the x_W and y_w are the two axis used for the angular motion calculation of the wrist.

3.4.1 Wrist Position Estimation

In this section, we will estimate the position of the wrist by utilizing the contour coordinate information extracted earlier. Additionally, we implement a minimum bounding rectangle algorithm around the arm contour in order to use its aspect ratio (width divided by the height of the rectangle) information to know the status of the arm at all time that is to know if the arm is extended or bent (Fig 19). The wrist position is estimated by identifying the two points on the arm contour that are near the wrist region (Fig 19).

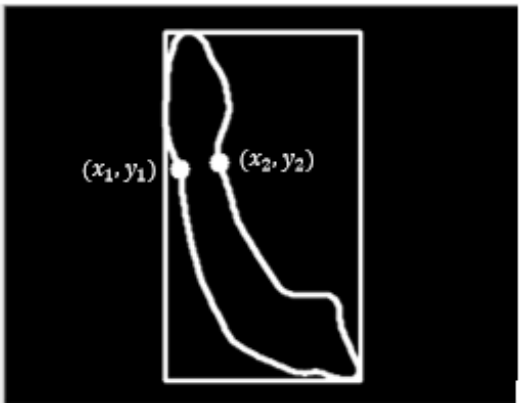


Fig 19: Points surrounding the wrist region.

Consequently, we compute the mid-point using the mid-point formula (Eq. 9) of the two identified points near the wrist to get an estimation of the wrist position. Fig 20 shows the estimated wrist position being overlaid on the arm.

$$\left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}\right) \tag{9}$$

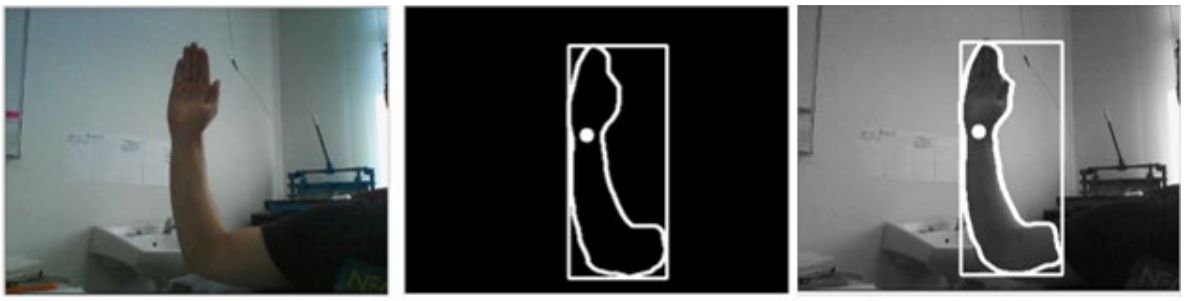


Fig 20: (a) example of arm; (b) the wrist position extraction; (c) the joint visualization.

Furthermore, we implemented a minimum bounding rectangle algorithm around the arms contour and use the aspect ratio of the rectangle to know the status of the arm (either flexion or extension). Knowing the status of the arm allows us to re-assign the two points near the wrist (Fig 21) differently based on if the arm is extended or bent so we could have a more precise estimation of the wrist position. Re-assigning the points is due to the redistribution of the contour points when the arm changes its position from being flexed to becoming extended. If the aspect ratio of the rectangle surrounding the arm contour is lesser than one it is considered up right (or bent) position. Fig 21 shows the two points near the wrist being correctly extracted regardless of the position of the arm.

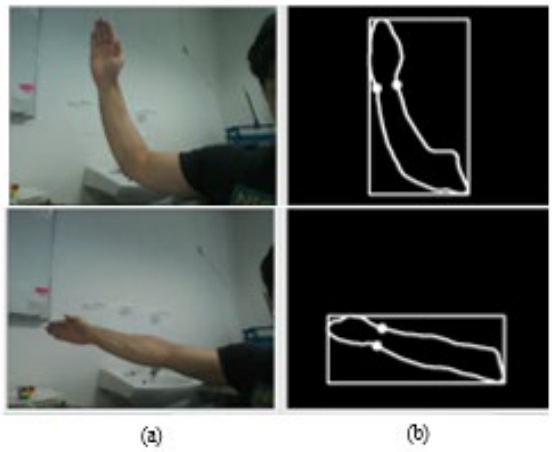


Fig 21: (a) Different arm positions; (b) correct extraction of two points near the wrist.

$$\frac{Rectangle\ width}{Rectangle\ height} < 1\,, \qquad (arm\ is\ not\ extended) \tag{10}$$

$$\frac{Rectangle\ width}{Rectangle\ height} > 1, \qquad (arm\ extended) \tag{11}$$

3.4.2 Elbow Position Estimation

To estimate the position of the elbow we apply the same method as we did for finding the wrist position. Fig 22 shows the estimated elbow and wrist position on the arm. Finally, we put all the information obtained to create a visual perspective of our arm model as shown in Fig 22 (c).

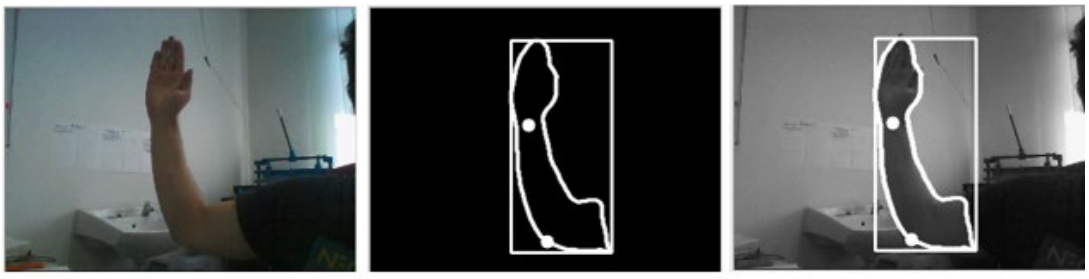


Fig 22: (a) example of arm; (b) the elbow position extraction; (c) the joint visualization.

Fig 23: (a) example of an arm; (b) wrist, elbow and shoulder joints extraction; (c) joints visualization.

Proposed wrist and elbow position estimation of the arm is based on DeLisa, Gans, and Walsh [24]. The wrist and elbow position points are our reference points for calculating the angular motion of the arm. To calculate the range of angular motions of the wrist and elbow we divided our arm model into two pairs of points as reference. The first pair of points are used for calculating the angular motion of the wrist as shown in Fig 23(a) and the second pair of points are used to calculate the angular motion of the elbow as shown in Fig 23(b).

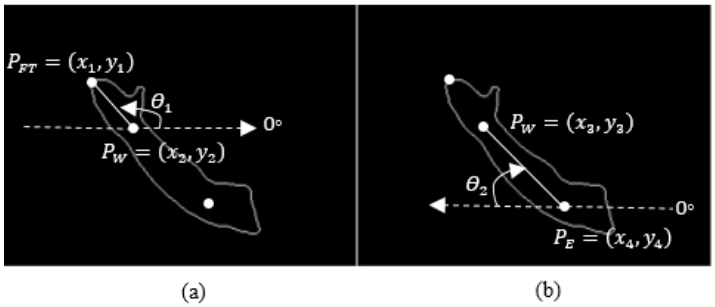


Fig 24: Range of angular motion calculation, for wrist and elbow respectively.

Positive (negative) rotation of the wrist around the x-axis by angle θ_1 corresponds to wrist flexion (extension) (Fig 24(a)). Positive (negative) rotation of the forearm around the x-axis by θ_2 corresponds to elbow flexion (extension) (Fig 24(b)).

$$\theta_1 = 2\arctan\frac{(y_1 - y_2)}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} + (x_1 - x_2)} \tag{12}$$

$$\theta_2 = 2\arctan\frac{(y_3 - y_4)}{\sqrt{(x_3 - x_4)^2 + (y_3 - y_4)^2} + (x_3 - x_4)} \tag{13}$$

4.0 EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, the performance of the proposed system is evaluated in two aspects. First, we evaluated the accuracy of a proposed vision-based measuring system for upper limb functions against conventional goniometer measuring system. The second test is to evaluate the performance of the proposed system with the external camera at different viewing angle with respect to goniometer.

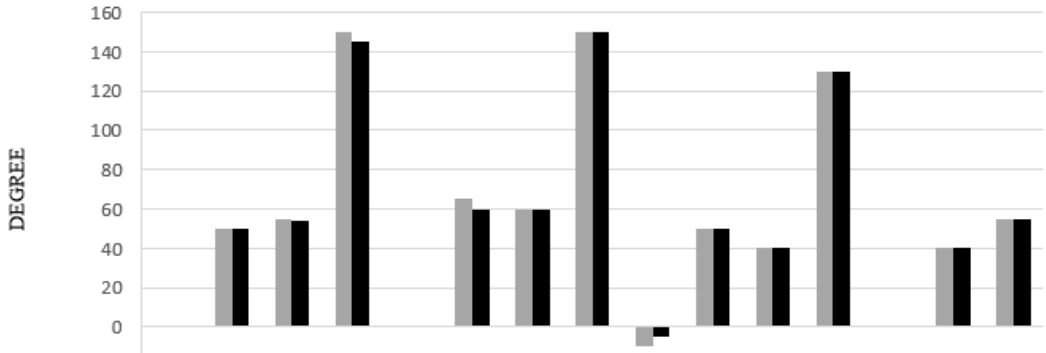
In the first setup, it measured the range of angular motions namely, “wrist extension”, “wrist flexion”, “elbow extension” and “elbow flexion” by a conventional goniometer. These are performed by the subjects as our ground truth data as shown in Fig 25. Furthermore, we repeated and measured the same range of movements from the participants in front of proposed vision-based measuring system (Fig 26). The proposed and conventional goniometer system based results are shown in Fig 27.

(a) (b) (c) (d)

Fig 25: Goniometer measurements (a) elbow extension; (b) elbow flexion; (c) wrist flexion; (d) wrist extension

(a) (b) (c) (d)

Fig 26: The measurements of proposed system (a) elbow extension; (b) elbow flexion; (c) wrist flexion; (d) wrist extension



	Elbow extension	Wrist flexion	Wrist extension	Elbow flexion	Elbow extension	Wrist flexion	Wrist extension	Elbow flexion	Elbow extension	Wrist flexion	Wrist extension	Elbow flexion	Elbow extension	Wrist flexion	Wrist extension
Goniometer	0	50	55	150	0	65	60	150	-10	50	40	130	0	40	55
Our system	0	50	54	145	0	60	60	150	-5	50	40	130	0	40	55

Fig 27: Comparison of measurements from a proposed system with a goniometer.

From Fig 27, it can be observed that proposed angular motion computation is almost accurate and close to the conventional measurements of goniometer by a professional physiotherapist. Each column on the table represents the value in degrees obtained from the goniometer and proposed system measurement. For example, the first column on the table shows a value of 0◦ for both goniometer and proposed system. This means when the subject performed an elbow extension exercise both the goniometer and our system produced the same reading. Furthermore, it can be observed the almost all the upper limb measuring (whether the elbow/wrist extension/flexion) bar values are almost similar in both goniometer and proposed system. Furthermore, Table 2 depicts the average error associated with each particular motion performed by the subjects. Where the maximum average error produced by the proposed system is around 1.25 degrees. Therefore, from above analysis results, propose measurement system has an acceptable accuracy and efficiency with very low average error (degree). In contrast, generally, the physiotherapists may not be able to always accurate to measure the range of angular motions using conventional tools such as the goniometer. So the proposed system can be considered as a more reliable angular measuring solution.

Table 2: The average error associated with each motion.

Range of Motions	Average error (Degree)
Elbow flexion	0
Elbow extension	1.25
Wrist flexion	1.25
Wrist extension	0.25

In the second test, we measure the performance as the error rate of the proposed system with the external camera at different viewing angles with respect to goniometer. We measured different angular rotations made by a goniometer at different viewing angles with respect to the position of our external camera. This indirectly shows that how many readings from the proposed system and the goniometers actually differ in term of errors. The results of this experiment will show that to what extent proposed system is able to produce accurate readings even with the external camera at different error based angles in the proposed system. We measured different sets of angles made by the goniometer with the camera placed right above it as shown in Fig 28. First, we considered the 0◦ degrees viewing angle, and repeated the same procedure for 20◦, 30◦ and 50◦ viewing angles of the camera with respect to the goniometer as depicted in Fig 29.

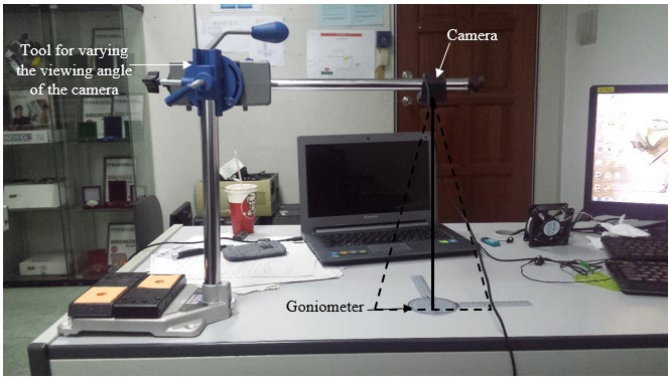
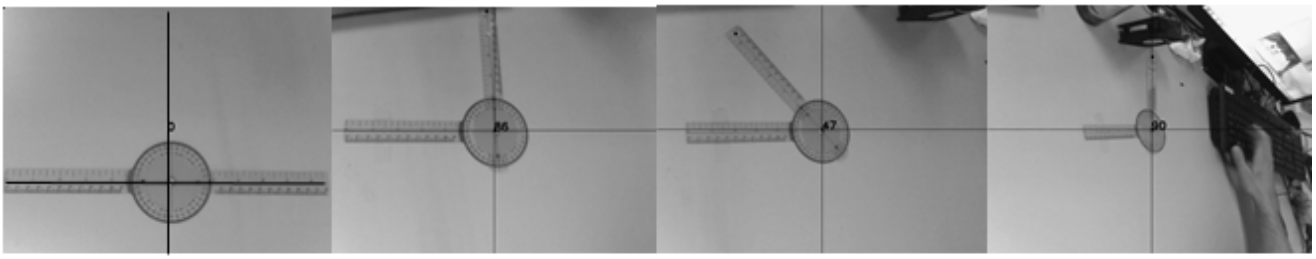


Fig 23: Different viewing angle experiment.



(a) (b) (c) (d)

Fig 29: Cameras different viewing angles; (a) camera at 0◦ to the goniometer; (b) camera at 20◦ to the goniometer; (c) camera at 30 ◦ to the goniometer; (d) camera at 50◦ to the goniometer.

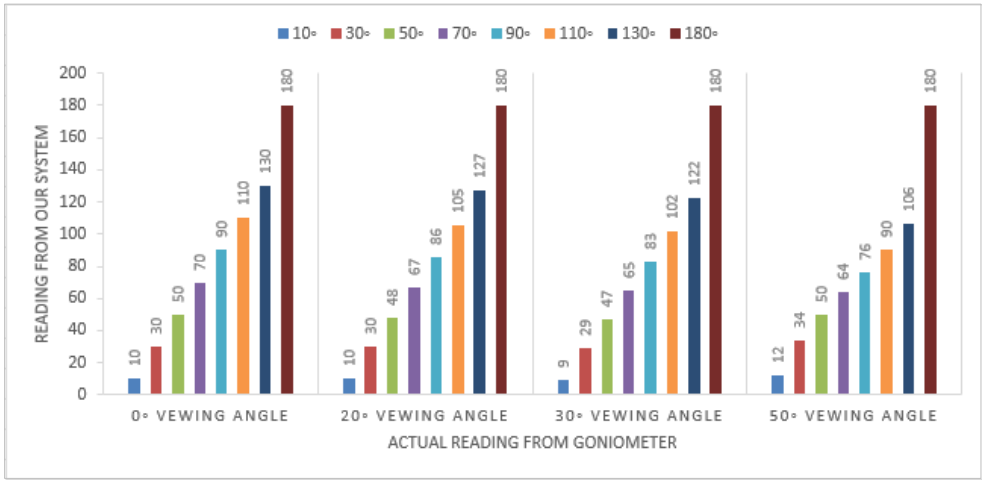


Fig 30: The comparison between proposed system angle evaluation and the actual angle at different camera point of view.

In Fig 30, it shows the comparison of the measurement made by the proposed vision-based system at the various point of view of a camera. Where, the 0◦ viewing angle column represents the different angles reading bars of the camera towards the goniometer. It showed that the proposed system can identical measures the angles once the goniometer has 0◦ angles. Once the viewing angles of a camera increased 20◦ to 50◦, the proposed system can measure slightly different values as compared to actual goniometer readings. For example, when we placed the camera at 50◦ angles towards the goniometer and the angle of goniometer is set to 110◦. In result, the proposed system computes the angle of 90◦ instead of 110◦, which shows the drift/error of 20◦ angle in the proposed system. However, this concludes that the proposed system is more reliable/accurate at lower point of view of cameras angles changes.

5.0 DISCUSSION

This paper present a single camera based, marker free upper limb measurement system for measuring elbow extension, elbow flexion, wrist flexion and wrist extension intended for home-based rehabilitation to monitor the progress of a stroke patients. A single camera and a marker free visual tracking system is proposed to reduce the cost of the system and to reduce the settings difficulty. The proposed contour based motion tracking is based on the coordinate points extracted from the contours of the object that able to overcome prior tracking techniques such Kalman filter, Optical flow and Cam-Shift. The proposed method overcome prior Kalman filter method by their simple design and fast computation procedure. As the method use predefined points, it able to mitigate Optical flow and Cam-Shift problems. Optical flow is based on intensity and Cam-Shift is based on histogram back propagation. Unlike the proposed predefined points, the two methods suffers from various lighting and color conditions that would change the intensity and histogram properties.

The contour based motion tracking is divided into skin segmentation and arm modelling for motion tracking. Our skin segmentation method adjusts the best values for the lower and upper threshold boundary of isolated skin regions from the background to overcome the problem of a fixed threshold value and color histogram adaptive thresholding. This is done to cope with different skin colors and different illumination conditions. We utilize the hue, saturation, and value (HSV) space which provides a high separation contrast between skin regions and non-skin regions. The proposed segmentation method is then compared with four other common skin segmentation methods such as skin probability map, HSV back-projection, YCrCb back-projection and RGB back-projection algorithm with the same image database (300 images from the MUCT face data base). The results show that our method has an accuracy of 93.2% and a true positive detection rate of 85.2%, which outperformed the prior methods. We also introduce a concept of filter truncating which applies several small bilateral filters rather than one large filter to allow faster processing time by discarding the minor parts of the filter as shown in Figure 8.

In the arm modelling for motion tracking, we formulate mathematical equation that model the points needed for the upper limb measurements based on the predictability of spacing of the points of the arm's contour. The contour based motion tracking, which combine the skin segmentation with the arm model is then compared with measurement taken by professional physiotherapist on a real human subject. The average error from the subjects for the elbow flexion was zero, meaning from all the subjects the elbow extension was correctly measured. The elbow extension measurement had an average error of 1.25 degrees, wrist flexion measurement had an average error of 1.25 degrees and finally the average error from the wrist extension measurement had a value of 0.25 degrees which overall shows the reliability of our system. A test between proposed system angle evaluation and the actual angle at different camera point of view is also conducted. The results shows that the error increases with the increase drift of viewing from the 0 degree angle. Therefore, the measurement should be taken at lower point of view of the viewing angles.

6. 0 CONCLUSION AND FUTURE WORK

Based on the results obtained, the proposed method has shown its capability to measure upper limb measurement needed by the physiotherapist. It produce an accurate and reliable reading of +-1.25 average range of error from actual physiotherapist reading. The home-based rehabilitation systems will enable both the patients and physiotherapist to track the effect of the suggested treatments. This method can be further extended to other measurements and to measure more than one subject. This is possible since camera vision has large angle of view that can be exploited. The progress of each individual within the camera view can be monitored more effectively.

ACKNOWLEDGMENTS

This work is funded by the University of Malaya Postgraduate Research Grant (PPP) (PG040-2015B) and Bantuan Khas Penyelidikan (BKP Special, ID 27483)

REFERENCES

[1] Mei, Z., Lin, L., Lei, Y., Lan, L., & Xianglin, F. *A Binocular Vision Motion Capture System Based on Passive Markers for Rehabilitation*. 2017. International Conference on Mechatronics and Intelligent Robotics (pp. 545-549). Springer, Cham.

[2] Zhou, H. and H. Hu, *Human motion tracking for rehabilitation—A survey.* Biomedical Signal Processing and Control, 2008. **3**(1): p. 1-18.

[3] Nichols-Larsen, D.S., et al., *Factors influencing stroke survivors' quality of life during subacute recovery.* Stroke, 2005. **36**(7): p. 1480-1484.

[4] Cirstea, M. and M. Levin, *Improvement of arm movement patterns and endpoint control depends on type of feedback during practice in stroke survivors.* Neurorehabilitation and neural repair, 2007. **21**(5): p. 398-411.

[5] Gómez-García, J.A., J.I. Godino-Llorente, and G. Castellanos-Dominguez, *Non uniform Embedding based on Relevance Analysis with reduced computational complexity: Application to the detection of pathologies from biosignal recordings.* Neurocomputing, 2014. **132**: p. 148-158.

[6] Liang, H. and T. Morie, *A motion detection model inspired by hippocampal function and its applications to obstacle detection.* Neurocomputing, 2014. **129**: p. 59-66.

[7] Choi, Y., S. Ozawa, and M. Lee, *Incremental two-dimensional kernel principal component analysis.* Neurocomputing, 2014. **134**: p. 280-288.

[8] Perez, P., J. Vermaak, and A. Blake, *Data fusion for visual tracking with particles.* Proceedings of the IEEE, 2004. **92**(3): p. 495-513.

[9]. Thomopoulos, S.C. and L. Nillson. *3-D motion tracking using stereo camera and range radar*. in *SC-DL tentative*. 1990. International Society for Optics and Photonics.

[10] Tao, Y. and H. Hu, *A novel sensing and data fusion system for 3-D arm motion tracking in telerehabilitation.* Instrumentation and Measurement, IEEE Transactions on, 2008. **57**(5): p. 1029-1040.

[11] Tao, Y., H. Hu, and H. Zhou, *Integration of vision and inertial sensors for 3D arm motion tracking in home-based rehabilitation.* The International Journal of Robotics Research, 2007. **26**(6): p. 607-624.

[12] Idris, M.Y.I., et al., *A co-processor design to accelerate sequential monocular SLAM EKF process.* Measurement, 2012. **45**(8): p. 2141-2152.

[13] Sucar, L.E., et al., *Gesture therapy: A vision-based system for arm rehabilitation after stroke*, in *Biomedical Engineering Systems and Technologies*. 2008, Springer. p. 531-540.

[14] Polana, R. and R. Nelson. *Low level recognition of human motion (or how to get your man without finding his body parts)*. in *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*. 1994. IEEE.

[15] Du, M., X. Nan, and L. Guan, *Monocular Human Motion Tracking by Using DE-MC Particle Filter.* Image Processing, IEEE Transactions on, 2013. **22**(10): p. 3852-3865.

[16] Ding, J., Z. Lin, and H. Xie, *A method of 3D recovery of human skeleton from monocular.* Management Innovation and Information Technology61, 2014. **379**.

[17] Liu, J., et al., *3D Human motion tracking by exemplar-based conditional particle filter.* Signal Processing, 2015. **110**: p. 164-177.

[18] Spruyt, V., A. Ledda, and W. Philips, *Robust arm and hand tracking by unsupervised context learning.* Sensors, 2014. **14**(7): p. 12023-12058.

[19] Adams, R.J., et al., *Assessing upper extremity motor function in practice of virtual activities of daily living.* Neural Systems and Rehabilitation Engineering, IEEE Transactions on, 2015. **23**(2): p. 287-296.

[20] Tian, Y., et al., *Upper limb motion tracking with the integration of IMU and Kinect.* Neurocomputing, 2015. **159**: p. 207-218.

[21] Tanaka, H., Y. Sumi, and Y. Matsumoto. *A portable 6-DOF motion tracker using high-accuracy AR markers—First report on the feasibility*. in *Machine Vision Applications (MVA), 2015 14th IAPR International Conference on*. 2015. IEEE.

[22] Ligorio, G. and A.M. Sabatini, *A novel Kalman filter for human motion tracking with an inertial-based dynamic inclinometer.* Biomedical Engineering, IEEE Transactions on, 2015. **62**(8): p. 2033-2043.

[23] Akhavizadegan, A. and M. Idris. *Camera Based Arm Motion Tracking for Stroke Rehabilitation Patients*. in *International Conference for Innovation in Biomedical Engineering and Life Sciences*. 2015. Springer.

[24] DeLisa, J.A., B.M. Gans, and N.E. Walsh, *Physical medicine and rehabilitation: principles and practice*. Vol. 1. 2005: Lippincott Williams & Wilkins.

[25] Ren, W., Chen, X., Wang, X., & Liu, M. *A Novel Optical Flow Algorithm Based on Bionic Features for Robust Tracing.* 2015 *In Proceedings of the 4th International Conference on Computer Engineering and Networks (pp. 669-675).* Springer International Publishing.

[26] Viola, P., & Jones, M. J. (2004). Robust real-time face detection. International journal of computer vision, 57(2), 137-154.

[27] Gomez, G., & Morales*, E. (2002, July). Automatic feature construction and a simple rule induction algorithm for skin detection. In Proc. of the ICML workshop on Machine Learning in Computer Vision (Vol. 31).*

[28] Soriano, M., Martinkauppi, B., Huovinen, S., & Laaksonen, M. *(2003). Adaptive skin color modeling using the skin locus for selecting training pixels. Pattern Recognition, 36(3), 681-690.*

[29] Yang, J., Lu, W., & Waibel, A*. (1998, January). Skin-color modeling and adaptation. In Asian Conference on Computer Vision (pp. 687-694). Springer, Berlin, Heidelberg.*

[30] Sirikuntamat, N., Satoh, S. I., & Chalidabhongse, T. H. *Vehicle tracking in low hue contrast based on CAMShift and background subtraction. In Computer Science and Software Engineering (JCSSE), 2015 12th International Joint Conference on (pp. 58-62). IEEE.*

[31] Biplab Ketan Chakraborty, M. K. Bhuyan, Sunil Kumar *A Weighted Skin Probability Map for skin color segmentation. In International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET),2016. IEEE*

[32] Kawulok M., Nalepa J., Kawulok J. (2014) *Skin Detection and Segmentation in Color Images. In: Celebi M., Smolka B. (eds) Advances in Low-Level Color Image Processing. Lecture Notes in Computational Vision and Biomechanics, vol 11. Springer, Dordrecht*

[33] Cheryl D. Metcalf, Rebecca Robinson, Adam J. Malpass, Tristan P. Bogle, Thomas A. Dell, Chris Harris, and Sara H. Demain (2013, August) *Markerless Motion Capture and Measurement of Hand Kinematics: Validation and Application to Home-Based Upper Limb Rehabilitation IEEE Transactions on Biomedical Engineering, Vol. 60, No. 8*