Ye, Minxiang and Yang, Cheng and Stankovic, Vladimir and Stankovic, Lina and Kerr, Andrew (2016) A depth camera motion analysis framework for tele-rehabilitation : motion capture and person-centric kinematics analysis. IEEE Journal on Selected Topics in Signal Processing. ISSN 1932-4553 (In Press) ,

A Depth Camera Motion Analysis Framework for Tele-rehabilitation: Motion Capture and Person-Centric Kinematics Analysis

[1]Minxiang Ye, [1]Cheng Yang, [1]Vladimir Stankovic, [1]Lina Stankovic, and [2]Andrew Kerr
[1]Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, G1 1XW, UK
[2]Biomedical Engineering Department, University of Strathclyde, Glasgow, G1 1XW, UK

***Abstract*—With increasing importance given to tele-rehabilitation, there is a growing need for accurate, low-cost, and portable motion capture systems that do not require specialist assessment venues. This paper proposes a novel framework for motion capture using only a single depth camera, which is portable and cost effective compared to most industry-standard optical systems, without compromising on accuracy. Novel signal processing and computer vision algorithms are proposed to determine motion patterns of interest from infrared and depth data. In order to demonstrate the proposed framework's suitability for rehabilitation, we developed a gait analysis application that depends on the underlying motion capture sub-system. Each subject's individual kinematics parameters, which are unique to that subject, are calculated and these are stored for monitoring individual progress of the clinical therapy. Experiments were conducted on 14 different subjects, 5 healthy and 9 stroke survivors. The results show very close agreement of the resulting relevant joint angles with a 12-camera based VICON system, a mean error of at most 1.75% in detecting gait events w.r.t the manually generated ground-truth, and significant performance improvements in terms of accuracy and execution time compared to a previous Kinect-based system.**

***Index Terms*—signal processing for rehabilitation, depth image processing, motion analysis, feature extraction, tele-rehabilitation**

I. Introduction

Following a stroke, the recovery of physical functions such as walking, could be greatly enhanced by the intervention of a rehabilitation team focused on the identification and resolution of movement problems, typically through the practice of exercise tasks. A range of movement abnormalities are periodically assessed to track and design rehabilitation progress for each individual patient. The outcome of rehabilitation is generally improved if the patient receives a high intensity of practice combined with feedback on their movement to correct errors [1]. However, this ideal type of therapy is restricted by access to professional rehabilitation staff and equipment, a situation which has led to the growing importance of self-management strategies, including the use of tele-rehabilitation.

Most motion analysis systems used for rehabilitation are based on multiple wearable sensors (e.g., passive/active optical markers, EMG/EEG/ECG, inertial sensors, force plates) and require a large laboratory space, are of high cost, and not portable, thus unsuitable for flexible, mobile clinical and home-use rehabilitation programs [2]. Optical motion analysis systems are attractive; however, current marker-based and marker-less, single or multiple infrared/RGB camera motion analysis systems have limitations, such as dependency on the underlying fabric color, time-consuming process, lack of portability and/or high price, such as VICON [3], single RGB camera systems of [4]–[7] and multiple RGB camera systems, such as [8]. Inertial tracker-based systems, like Xsens MVN [9] and M3D [10], are options for large clinics or hospitals, but are not suitable for small clinics and home use.

Alternatively, single RGB-depth camera systems, such as [11], [12], [13], after significant technological advances, have become cheap and popular options. For example, Microsoft (MS) Kinect enables tracking of human joints in three dimensional (3D) space using a single camera and its SDK via skeleton tracking [11]. However, Kinect's skeleton data are too noisy (see, e.g., Fig.1 in [14]), and do not provide sufficient accuracy [14]–[17]. Using two Kinect sensors, as in [18], can potentially improve the accuracy, but at the expense of portability, required expertise, and ease of setup.

The marker-less Kinect-based approach of [19], for performing the 'Get Up and Go Test', which is part of the larger Tinetti test to identify subjects at risk of falling, is based on the construction of the background depth frame, which enables background removal, followed by frontal pose analysis to get body structure parameters and the sagittal view joint trajectory estimation. The method does not achieve clinical accuracy showing an error of up to 15 pixels compared to the reference trajectory. Six joints are tracked in the sagittal plane; the foot joint was not tracked, and it is not expected to work well due to interference with the floor. A similar approach [20] uses RGB and depth images of MS Kinect for semi-automatic postural and spinal analysis using Dynamic Time Warping, pose estimation and gesture recognition. The algorithm requires substantial manual effort, operation expertise and is time-consuming, hence not suitable for real-time application. Note that [19] and [20] are not validated against state-of-the-art benchmarks. [21] uses Kinect's depth images to perform 3D pose estimation with high computational complexity and is unsuitable for near real-time processing. [22] relies on Kinect SDKs virtual skeleton of the body and supervised learning to extract positions of the joints of interest in a gait analysis application, but is limited by high computational complexity, need for training data, and presents no scientific evidence that the proposed methods are clinically accurate. [23] uses two cameras and requires complex calibration, camera synchronization and setup.

In this paper, we develop a general framework to facilitate the next generation of portable and cost-effective telerehabilitation applications, suitable for local clinics and home use, that do not require any clinical expertise to operate. The proposed framework combines high accuracy marker-based tracking methodology based on infrared (IR) sensing and portability and affordability of range imaging methodology using structured light or Time-of-Flight sensors. Our proposed kinematics framework is capable of building various motion analysis assessment tools that target different rehabilitation applications. In contrast to previous work [19] and [20], our proposed framework is benchmarked against the state-of-the-art gold standard optical motion system VICON [3] for gait analysis using the walk forward and back test, with 6 markers on each sagittal plane (left and right) to capture both sagittal planes during the walking test in one go, and most importantly, create a person-centric subject model to define the geometric relationship between different markers. Additionally, as opposed to [6], [7], [19], our framework maps markers in 3D space since 2D measurements are nonlinear due to the fish eye effect from the sensor lens; the depth information for the marker centroid in the depth hole is recovered to perform coordinate mapping from image space to camera space.

The framework is based on several image processing algorithms, that enable extraction of specific movement patterns from IR and depth image data, is robust to occlusion, and facilitates real-time post-processing and visualization of the results. Namely, the main contributions of the paper are (see Section II for more details): (1) Single-camera imaging methodology, including scene calibration and denoising, where only one IR-based depth camera is used for motion capture, (2) simultaneous marker detection and identification in 3D space using adaptive thresholding with a novel depth recovery method to map the object coordinates into camera space, (3) person-centric model-based kinematics analysis, including effective post-processing motion analysis algorithms.

We provide detailed algorithmic steps for the proposed algorithms, making the proposed approach reproducible. The paper is organized as follows. In Sec.II, the overall description of the proposed framework is given followed by detailed descriptions of the proposed optical motion capture system and kinematics analysis algorithms in Secs. III and IV, respectively. Sec.V presents our visualization tools and experimental results, before concluding in Sec.VI.

II. Overview of the Proposed Framework

The proposed framework comprises an optical motion capture system and kinematics analysis tools that enable secondary development for solution enhancement. The interconnection among the underlying algorithms and key parameters used in the algorithms listed in this section.

The optical motion capture system (described in Sec.III) consists of a single depth camera (both IR and depth images are used) that enables creating 3D optical motion reconstruction. It is designed to capture human motion in real time by detecting retro-reflective markers attached to joints of interest, and comprises three modules: (1) Data cleaning - for cleaning IR and depth images (described in Sec.III-A). (2) Detection - for tracking markers in image space (Sec.III-B). (3) Mapping - for recovering the markers' position in camera space through the proposed cluster location algorithm (Sec.III-C).

The proposed kinematics analysis tools are developed as an application solution that sits on the proposed motion capture system, facilitating portable, indoor tele-rehabilitation diagnosis, as demonstrated by our gait analysis application in Sec.IV. Autonomously located markers attached to subject's joints during the straight-line walking exercise, are used to automatically calculate gait associated parameters commonly used for clinical assessment, such as joint angles, velocity, movement patterns, gait cycle phase, step and stride length, swing and stance phase, etc. [24]. In particular, the gait analysis tools (Sec.IV) comprise: (1) Scene calibration (Sec.IV-A), (2) Subject modelling (Sec.IV-B) for building a person-specific body segmentation model, (3) Kinematics analysis module (Sec.IV-C) for calculating gait analysis parameters based on the proposed analytics.

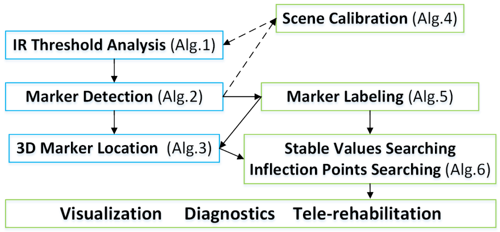

Scene Calibration (Alg.4)
IR Threshold Analysis (Alg.1)
Marker Detection (Alg.2)
Marker Labeling (Alg.5)
3D Marker Location (Alg.3)
Stable Values Searching
Inflection Points Searching (Alg.6)
Visualization
Diagnostics
Tele-rehabilitation

Fig. 1. Overall proposed system structure diagram.

During processing of the IR images, we observed that motion blur and light conditions strongly influence the speed and accuracy of marker tracking. Moreover, since the retro-reflective markers block the depth measurements from the depth camera, the only way to recover the depth value for each marker is to use their surrounding information. To address the above problems, we proposed three algorithms: (1) Threshold analysis (Alg.1) – extending previous work in [25] to solve fast motion and camera noise during marker detection, (2) Marker detection (Alg.2) – the idea is to improve the marker centroid location accuracy and speed which are attached to joints of interest, in image space, (3) Depth recovery and mapping (Alg.3) – the 3D texture is partially missing in the marker region and it is possible to use the point cloud histograms for restoring the depth value of the marker centroid. When looking at the point cloud histograms, we can get a kernel that has higher weight inside according to their Euclidean distance to the marker centroid and frequency of occurrence.

Once the coordinates of the markers in 3D space have been obtained above, the aim of the gait analysis application is to label or associate the markers to joints on the human subject, so that joint angles can be calculated during kinematics analysis. In order to do so, the first step is Scene calibration

(Alg.4) whose purpose is to map the physical measurements of the physical experimental environment into a virtual environment, recreating a geometric relationship between the camera, calibration markers, and walking start/end points. This enables Marker labelling (Alg.5), where a person-centric subject model is constructed to map the subjects physical dimensions to virtual 3D space building a geometric relationship among markers on the body and hence markers can be accurately labelled as belonging to the foot joint, hip, etc. Once all the markers have been labelled, kinematics analysis commences in Gait event detection (Alg.6) by examining the relative trajectories of knee, ankle, and heel markers to the floor (obtained during scene calibration and marker labelling) to find inflection points and local peaks for gait events detection without pre-smoothing the data, in order to get the best accuracy, including addressing occluded markers.

Table I lists key parameters used in the six proposed algorithms, and which values were observed to give the best results when trading accuracy and execution time. Note that we observed that changes in the values of the parameters result in small changes in the accuracy of results and execution time. These pre-set values were selected based on the MS Kinect 2 sensor, a walking line distance to the camera of 2.5m-3m and sensor height of about 0.8m from the floor, resulting in an approximate 4m walking line. If the camera sensor and the latter distances are changed, a standard calibration procedure (e.g., [7], [26]) can be used to find optimal values for D_0, D_1 and D_2.

III. Optical Motion Capture System

The task of the proposed optical motion capture system is to simultaneously track multiple retro-reflective markers using a single IR depth camera, irrespective of the overlying motion analysis application. Retro-reflective materials were chosen since they introduce high intensity regions into IR images and blank holes into depth images. Therefore, the markers are detected on IR images, after which the marker location is recovered in the depth image and mapped to camera space via the following key steps: (1) Data cleaning - cleaning invalid data and reducing sensor noise, (2) Marker detection - detecting markers in IR image space using connected component algorithm [27] with scene dependent adaptive thresholds, (3) 3D marker location - recovering marker depth values using our novel cluster location algorithm (in Sec.III-C) and mapping depth space coordinates to the camera space using the depth-map projection method of [28]. We elaborate each of these steps in the following subsections.

A. Data cleaning

The primary source of noise affecting the captured IR images is from the camera lens of either IR transmitters or receivers and interfering sources such as metallic materials, retro-reflective materials, etc. Reflective materials other than markers will influence the measurements and constitute interference while recovering depth values. Fig.2 shows the noise, originating from the imaging sensor and reflective material, typically encountered in an acquired frame.

TABLE I
Parameters used in the proposed algorithms

Param.	Alg.	Description	How is it set?
S	$1, 4$	Captured infrared image sequence	Measured by the sensor
w	1	Blob detection threshold	Initialised in Alg.4 and updated in Alg.1
r_b		Blob base radius	Calculated by Alg.4
n		Number of markers used	Application specific (12 in the experiments)
b		Scan window length of the previous frames	3 frames (heuristically)
S_d	2	Captured current infrared image	Measured by the sensor
D_d		Captured current depth image	Measured by the sensor
(p, q)	3,5	Marker centroid coordinates	Calculated by Alg.2
r_m		Marker radius	
r_r		Marker region radius	
W	3	Max-Min width	50 mm (heuristically)
D_0		Recovery resolution	2 pixels (heuristically)
D_1		Depth resolution (Sensor accuracy)	5 mm (heuristically)
D_2		Distance resolution	0.5 mm (heuristically)
m		Cluster mode	Application specific
C	4	Number of calibration markers	Application specific
γ	6	Level resolution	0.05 (heuristically)
ξ		Range left clip rate	0.03 (heuristically)
ϕ		Range right clip rate	0.03 (heuristically)
τ		Local range length boundary	3 (heuristically)

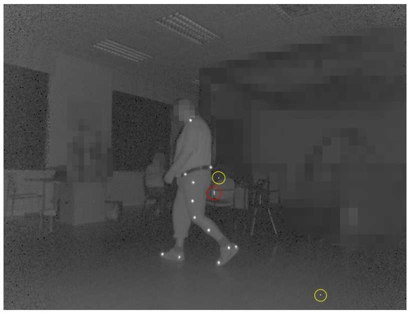


Fig. 2. Noise from the sensor (yellow) and reflective material (red)

Approaches for denoising include depth map denoising, either spatially with, e.g., adaptive total variation [29], nonlocal graph-based transform with group sparsity [30] and layer-based depth correction and completion [31], or temporally with, e.g., parametric model-based nonlocal means [32] and joint-bilateral filter [33]. Such data cleaning approaches would potentially preserve sharp edges without over-smoothing, and improve the accuracy of marker tracking. However, since we are aiming for near real-time applications, we use a simpler, intuitive and less complex, but effective approach based on Kalman filtering [34]. Namely, since we can detect initial locations of interfering materials in the first frame and corresponding pixel values, it is easy to predict their next state using Kalman filter, and exclude them from further processing.

B. Detection

After cleaning the frame from unwanted noise, IR images are converted to binary format in order to detect and identify markers via blob detection on a frame-by-frame basis. Since all retro-reflective marker regions have clearly distingushable pixel values in IR images from surounding regions, blob detection is a natural object detection choice.

There are several approaches to detect and identify blobs, such as matched filters / template matching [25], watershed detection [35], structure tensor analysis followed by hypothesis testing of gradient directions [36], [37], scale-space analysis [38]. All these approaches are limited by their sensitivity to noise, structure restriction and complexity [39]. In previous related work [25], a concentric cycle-based method (template matching) is proposed to perform the shape fitting test for each potential blob in order to locate all markers in image space (2D); however, this method is time consuming and requires expertise to determine associated parameters for the shape fitter and the kernel cluster filter, and cannot locate the center of the marker correctly when motion blur occurs and the marker is out of the sagittal plane, which leads to center deviation on those markers with circular distributed IR values.

To solve this problem and satisfy our real-time processing constraints, an enhanced heuristic IR analysis algorithm is proposed in Alg.1, where the threshold value is adaptively acquired for blob detection in the next frame. A sequence of b previous IR images and an initial threshold for blob detection w and blob base radius r_b, obtained by Alg.4 during the scene calibration process (which is application and scene dependent), are fed to Alg.1. Note that the initial threshold w in Alg.1 has little influence on the accuracy of the adaptive IR threshold. As expected, the further away the value of w from the optimal value, the higher the number of iterations to find a suitable threshold, resulting in longer execution time. Note that we stop iterating when the number of detected blobs f reaches convergence, i.e., the value of f between iterations is unchanged.

The main idea behind Alg.1 is to first assign w, for the current frame S_d to that used in one of the b previous frames, which results in the number of blobs in S_d closest to the actual number of markers n present in the scene. If this threshold detects more than n blobs (that is, some detected blobs are not markers), w is calculated by averaging the pixels from the n most significant blobs weighted by the their radius. Otherwise, if some markers were missed, a weighted average is taken over all detected blobs in the current and b previous frames.

After blob detection threshold is set, we use the connected component labelling algorithm [27], a classic blob extraction method used to detected connected regions in a binary image, to detect markers from the located blobs. Then, for each detected marker we find a centroid, radius and region radius using simple pixel-based geometry. The overall proposed algorithm is detailed in Alg.2.

Algorithm 1: Adaptive blob detection threshold setting for the next frame

```
Input: Captured image sequence from the sensor, $S$;
       Initial blob detection threshold, $w$ from Alg.4;
       Blob base radius, $r_b$ from Alg.4;
       Number of markers used, $n$;
       Scan window of $b$ previous frames
       $S_{d-b}, \dots, S_{d-1}$, with their blob detection
       thresholds $e_j, j = d-b, \dots, d-1$;
Output: Blob detection threshold for the next frame, $e_d$;
1  acquire next IR frame $S_d$ from $S$;
2  set $e_d = e_{j^*}$, where $j^* = \arg\min_{j=d-b,\dots,d-1} |n - f_j|$,
     where $f_j$ is the number of blobs detected in Frame $S_d$
     when blob threshold $e_j$ from Frame $S_j$ is used;
3  set $f$ as the number of detected blobs when $e_d$ is used
     on Frame $S_d$;
4  order the detected blobs into the descending order of the
     blob radius: $i_1^d, \dots, i_n^d, \dots, i_f^d$, where $d$ denotes the
     frame number;
5  set $J_q^d, q = 1, \dots, f$ as a matrix of all IR pixel values in
     Blob $i_q^d$, $k_q^d$ their mean value, $l_q^d$ and $u_q^d$ as radius and
     blob region radius of Blob $i_q^d$ in $S_d$, respectively;
6  if $f > n$ then
7      calculate new $e_d$ by averaging IR pixel values from
         $J_1^d, \dots, J_n^d$ weighted by $l_1^d, \dots, l_n^d$;
8  else if $f < n$ then
9      set $h_0$ by averaging IR pixel values from $J_1^d, \dots, J_f^d$
         weighted by $l_1^d/r_b, \dots, l_f^d/r_b$
10     set $h_1$ by averaging IR pixel values from
         $J_1^q, q = d-b, \dots, d-1$ weighted by $e_q/w$
11     set $h_2$ by averaging blob radius from $u_1^q$,
         $q = d-b, \dots, d-1$ weighted by $e_q/w$
         $e_d \Leftarrow (h_0 * f/n + h_1 * h_2/r_b + w)/(f/n + h_2/r_b + 1)$;
12 if $f_{last} \neq f_{current} \neq n$ then
13     add $S_d$ to scan window when using $e_d$ and goto 4;
14 return $e_d$;
```

C. 3D marker location

Once all blobs have been detected as valid markers, the next step is to obtain the coordinates of the markers in 3D space. In general, a depth camera has intrinsic parameters to perform spatial mapping from image space to camera space.

The depth-map projection method of [28] is adopted to acquire undistorted camera space coordinates of the tracked markers after marker centroids have been located. However, depth information within the marker region is empty due to the retro-reflective nature of the attached markers. Therefore, we propose to recover the sensitive pixels around each marker region in the depth images in Alg.3 by calculating image histograms with respect to pixel intensity (Steps 21 to 28 in Alg.3) and distance to the marker centroid in IR images (Steps 29-32). The algorithm is executed for each detected marker. The following parameters are assigned heuristically to improve the recovery accuracy and are constant for all frames: Max-Min width, $W = 50$, recovery resolution, $D_0 = 2$, histogram

```
Algorithm 2: Marker Detection
   Input: Captured IR image frame, $S_d$;
          IR blob detection threshold, $e_d$ obtained by Alg.1;
   Output: Marker centroid, $(p,q)_1, \ldots, (p,q)_n$ where $n$ is
           the number of detected markers;
   Marker radius, $r_{m_1}, \ldots, r_{m_n}$;
   Marker region radius, $r_{r_1}, \ldots, r_{r_n}$;
 1 Use connected component labelling [27] on $S_d$ with $e_d$
     for IR-to-binary image conversion and obtain labelled
     markers $M_1, \ldots, M_n$;
 2 foreach marker $M_i$ in $M_1, \ldots, M_n$ do
 3     set $\varrho$ as the number of pixels in $M_i$;
 4     set $g$ as the sum of all IR pixel values in $M_i$;
 5     set $v = g - \varrho * e_d$ as normalized sum of IR values;
 6     let $(p_i, q_i) = (0,0)$, $r_{m_i} = 0, r_{r_i} = 0$ be $M_i$'s
         centroid, radius and region radius, respectively;
       foreach pixel $P_{x,y}$ in $M_i$ do
 7         $p_i = p_i + x * P_{x,y}/v$; $q_i = q_i + y * P_{x,y}/v$;
 8     foreach pixel $P_{x,y}$ in $M_i$ with coordinates $(x,y)$ do
 9         set pixel distance $l = \sqrt{((x-p_i)^2 + (y-q_i)^2)}$;
10         $r_{m_i} = r_{m_i} + l * P_{x,y}/v$;
11         if $r_{m_i} > r_{r_i}$ then $r_{r_i} = r_{m_i}$;
12 return $(p,q)_1, \ldots, (p,q)_n$, $r_{m_1}, \ldots, r_{m_n}$, $r_{r_1}, \ldots, r_{r_n}$;
```

```
Algorithm 3: 3D marker Location
   Input: Captured depth image frame, $D_d$;
          Marker centroid, $(p,q)$, Marker radius $r_m$ and
          region radius $r_r$ obtained by Alg.2;
          Max-Min width, $W$;
          Recovery resolution, $D_0$;
          Depth resolution, $D_1$;
          Distance resolution, $D_2$;
          Cluster mode, $m$ [defined in Sec. III-C];
   Output: Marker position in frame $D_d$, $(x,y,z)$;
 1 acquire depth values $V^d$ at rectangle region of
     $\{$left:$p - r_r - D_0$, top: $q - r_r - D_0$, right: $p + r_r + D_0$,
     bottom: $q + r_r + D_0\}$ in $D_d$;
 2 order pixels in $V^d$ in the increasing order
     $v^d(1), \ldots, v^d(N)$, and set $\kappa = 0$ and $z = 0$;
 3 set $\Lambda_0 = v^d(1) + W$;
 4 set $V_0$ as a vector of all depth values in $V^d$ smaller than
     $\Lambda_0$ and the remaining values as $V_2$ , and set $V_1 = V_0$;
 5 if sizeof$(V_0) > 2$ then
 6     let $\kappa = \kappa + 1$, $V_0 = V^d \setminus V_0$ and goto 5;
 7 else if sizeof $(V_0) = 1$ then
 8     set $\Lambda_1 = v^d(N) + W$; set $V_1$ as all depth values in
         $V^d$ smaller than $\Lambda_1$ and the remaining values as $V_2$;
         goto 6 when sizeof$(V_2)$ >sizeof$(V_1) + \kappa$, otherwise
         goto 13;
 9 else
10     if $m =$ normal then
11         set $T_0 =$min$(V_0)$ and $T_1 = T_0 + W$;
12     else if $m =$top then
13         set $T_0 =$min$(V_1)$ and $T_1 = T_0 + W$;
14     else if $m =$bottom then
15         set $T_1 =$max$(V_2)$ and $T_0 = T_1 - W$;
16 set $H_0$ as the histogram of pixels in $V^d$ that fall between
     $T_0$ and $T_1$, with depth resolution $D_1$;
17 foreach bin $h$ in $H_0$ do
18     if sizeof$(h)$ <min$((r + D_0)^2, d^2/$sizeof$(H_0) + D_0)$
         then
19         if sizeof$(h)$ <min$(D_0,$sizeof$(v^d)/$sizeof$(H_0))$
             then
20             remove $h$ from $H_0$;
21 foreach $h$ in $H_0$ do
22     histogram all pixels in $h$ w.r.t their distance to
         centroid $(p,q)$, with bin resolution $D_2$;
23     set $\epsilon(h)$ as the mean value of the bin in $h$ that has
         the highest count;
24 foreach $h$ in $H_0$ do
25     set $z = z + \epsilon(h)*$sizeof$(h)/$sizeof$(H_0)$;
26 return $(x,y,z)$ mapped from $(p,q,z)$ using [28];
```

depth resolution, $D_1 = 5$, histogram distance resolution, $D_2 = 0.5$.

Alg.3 tackles the problem of partial occlusion: the input to the algorithm is a cluster mode variable, m that can take 3 possible discrete values: (1) Normal - no occlusion of the marker, (2) Top - occlusion present at the top of the marker, (3) Bottom - occlusion at the bottom of the marker. Partial occlusion takes place on markers attached to the anterior superior iliac spine (ASIS), posterior superior iliac spine (PSIS), hip and femur during arm swing. Those markers are in the bottom mode, while heel, toe, shoulder markers are in the top mode and the remaining markers are always in the normal mode in Alg.3 since they are never occluded. The proposed algorithm recovers depth information for each labelled marker independently even when partial occlusion occurs.

IV. Gait Analysis Application

This section describes the proposed application-specific algorithms that interface with our motion capture system (see Sec.III). The proposed gait analysis application, comprising scene dependent calibration, person-centric modelling, and kinematics analysis, enables autonomous, high-accuracy processing of gait associated data. Each of the three algorithms are explained next.

A. Straight-line Walking Scene Calibration

The purpose of scene calibration is to collect scene dimensions to build a geometric relationship between the camera, calibration markers, and walking start/end points.

A typical straight-line walking exercise scene captured by the camera, is represented as a virtual trapezoidal cylindrical model in Fig.3. The plane defined by 4 optical (calibration) markers, shown as blue dots in Fig.3, placed on the ground is perpendicular to the plane defined by the camera and the ground. An example of an IR image captured during

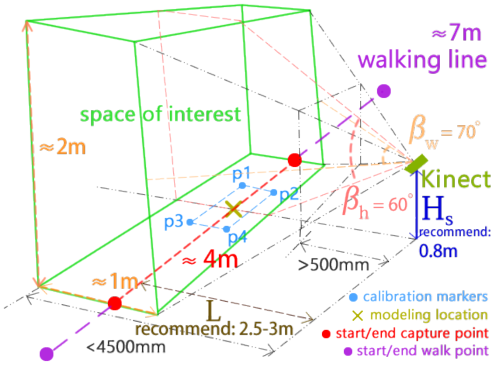


Fig. 3. Virtual straight walking exercise scene [25].

the calibration is shown in Fig.4, where the start and end of walking are shown as red dots. Previous experiments, as validated using a method of [26], showed that a walking line distance of $2.5-3$m to the camera and the sensor height of 0.8m from the floor result in an approximate 4m walking line.

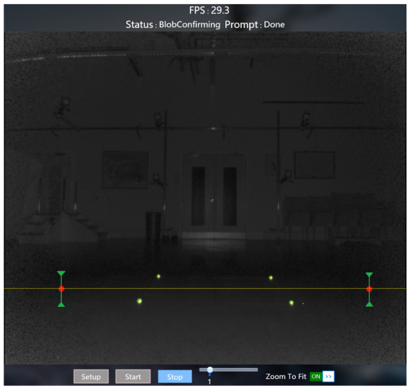

FPS : 29.3
Status : BlobConfirming Prompt : Done

Fig. 4. Scene calibration: Calibration markers in green are labelled (sagittal view). Walking line is defined between the start and end points, shown in red.

The overall walking scene calibration process is summarized in Alg.4. $C = 4$ calibration markers are placed on the ground one-by-one. The scene calibration process continuously searches and analyzes the status of the calibration marker plane in relation to the camera to ensure perpendicularity, and reports marker status as: (1) Uninitialized - stop mode, (2) Move Left/Right - camera needs to be moved to the left or right, (3) Tilt Down/Up, (4) Pan Left/Right, (5) Replace Markers - critical noise detected or marker placement error, (6) Done - calibration completed. Steps 12-15 perform manual adjustment of the camera pose.

Threshold w for blob detection (used in Alg.1) is calculated by first forming a histogram of edge pixels for each detected blob, and then finding the minimum (over all four marker blobs) of the largest histogram bin (Step 12). Alg.4 relies on subtracting the background to label the calibration markers and calls Alg.2, with updated w set to the minimal pixel value in the detected blob, to obtain the calibration marker's centroid and corresponding blob radius. Base blob radius r_b is set as the mean radius of all calibration markers. Alg.4 determines the start and end points of the walking exercise, which are then physically marked on the floor using a tape.

Algorithm 4: Gait Analysis Walking Scene Calibration

```
Input: Captured IR image sequence from the sensor, $S$;
       Number of calibration markers, $C$;
Output: Blob centroids, $(p,q,z)_1,\dots,(p,q,z)_C$;
        Start and end walking point, $(r,s)_0,(r,s)_1$;
        Initial IR blob threshold, $w$;
        Blob base radius, $r_b$;
        Walking line length, $L$;
1  set the number of labelled markers $c = 0$;
2  while $c \leq C$ do
3      repeat acquire the next IR image from $S$;
4      apply frame subtraction detection;
5      until no significant motion detected;
6      apply frame subtraction detection using as
         background the previous frame with no motion
         detected;
7      if blob detected then
8          update markers' state using marker labelling
             (call Alg.2 with $e_d$ set to the min IR value in
             the marker blob), and let $c = c + 1$;
9  calculate a histogram of edge pixel values for each blob,
     and set $w$ as the minimum, over all blobs, of the most
     significant bin.
10 check diagonal connection condition for
     $(p,q,z)_1,\dots,(p,q,z)_C$ mapped using normal mode
     Alg.3 with current depth image from $S$;
11 if connection is intersectant then
12     report plane status defined by $(p,q)_1,\dots,(p,q)_C$
         relative to camera;
13 else
14     report critical error and goto 1;
15 adjust camera's pose according to the reported status;
16 set $r_b$ =mean$\{r_1,\dots,r_C\}$, where $\{r_1,\dots,r_C\}$ are
     obtained by Alg.2 called in Step 8 above;
17 def start/end points $(r,s)_0,(r,s)_1$ relative to center of
     $(p,q)_1,\dots,(p,q)_C$ during streaming with guideline tool;
18 calculate the distance between $(r,s)_0$ and $(r,s)_1$ in
     camera space as $L$;
19 return IR base threshold $w$, blob base radius, $r_b$ walking
     line length $L$ and visualize start and end points
     $(r,s)_0,(r,s)_1$ in IR/RGB stream.
```

B. Model

Following calibration of the experimental environment, a unique complete subject model for sagittal gait analysis is constructed for every individual subject by physically measuring the subject standing at the location shown as X in Fig.3, specifically measuring $H0^*$, $H7^*$, and $W3^*$ to $W9^*$ (as shown in Fig.5) after all markers have been mapped in 3D space. The model is clustered into three parts: upper body, limb and foot models shown in Fig.5. For each frame, the model comprises the following: (i) position of all detected markers, (2) geometric relationship between markers, (3) virtual lines L13-L16 relative to the marker positions.

Each marker is labelled by examining all potential marker

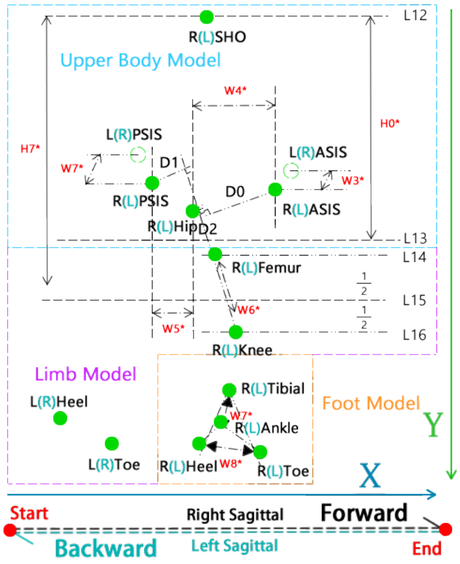


Fig. 5. Sagittal Model. 12 visible markers are marked with green circles. 2 partial invisible markers are shown in circle outlines. 'R' ('L') denotes right (left) marker. For example, RPSIS is the right posterior superior iliac spine marker [25].

groups for upper body, limb and foot models using Alg.5, scanning each IR frame from left-top to right-bottom. In particular, the shoulder (SHO) marker is first chosen as the top marker in the first frame that shows the subject in the sagittal plane and is labelled within the region around virtual line $L12$, predicted by Kalman filtering [34] using the marker position in the previous frames and its velocity. Then, all relevant distances (see Fig.5) are updated using the subject model of the previous/reference frame in order to solve model matching errors due to complete occlusion occurences on hip and femur markers. Once geometric relationships (distances, locations) relative to virtual lines $L12 - L16$ are determined, geometric relationships between all marker combinations will be checked. For example, the upper body model marker group should satisfy $D0 > D1 > D2$, $X_{LPSIS} < X_{RPSIS} < X_{RHIP} < X_{RASIS} < X_{LASIS}$, and the ankle marker of the foot model should be inside the triangle region defined by tibial, toe, heel markers. Potential clusters are formed by calculating the distances between the markers in the cluster and comparing them with the updated distances $W3^*$ to $W9^*$.

Since the geometric location relationship of the limb model markers is changing along the Y axis during leg swing, we select six markers on the bottom of the model along the Y axis and determine the two heel and toe markers attached to the occluded body side, by their relative position to the knee and other visible markers. Finally, the marker name/position is determined by comparing the distances between the markers in each cluster relative to the updated distances $W3^*$ to $W9^*$ for each validated marker cluster across the three sub-models.

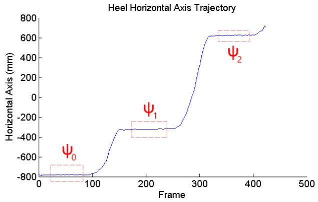
C. Kinematics Analysis

Once all the markers have been labelled, kinematics analysis commences, closely following the relative joint angle and gait cycle definitions from [24]. The relative trajectories of knee, ankle, and heel markers to the floor are examined to detect the following gait phases: initial contact, loading response, mid stance, terminal stance, pre-swing, initial swing, mid swing and terminal swing.

Algorithm 5: Marker Labelling

```
Input: From Alg.2:
        Centroids for $n$ markers, $(p,q)_1, \ldots, (p,q)_n$;
        Markers' radius, $r_{m_1}, \ldots, r_{m_n}$;
        Markers' region radius, $r_{r_1}, \ldots, r_{r_n}$;
        Marker positions in the previous frame, $F$;
Output: Labelled/named markers
predict SHO marker from $F$ using Kalman Filter [34];
if SHO not found then
    set centroid of the predicted region as SHO marker
      with radius and region radius as in $F$;
calculate all $W$'s and $L$'s values shown in Fig.5 using
  the current model (see Subsection IV-B);
order all markers in the region of L12 and L13 by
  X-coordinate;
determine the most-likely marker cluster for upper body
  based on D0, D1, D2 (see Subsection IV-B);
order markers under L13 by Y, and X afterwards.
divide lower limb markers into two clusters by
  evaluating 6 markers nearest to the ground by testing all
  possible clusters for the triangle foot model.
combine markers on the other side of the body into the
  triangle foot model in the upper limb region according
  to Y-coordinates;
determine the other side's foot position by checking its
  relative position with knee and foot marker;
map labelled $(p,q)_1, \ldots, (p,q)_n$ using Alg.3 with
  $r_{m_1}, \ldots, r_{m_n}$ and $r_{r_1}, \ldots, r_{r_n}$;
return labelled/named markers;
```

If a marker is occluded (full marker occlusion happens occasionally on the hip and femur markers), we adopt the 2nd or 4th cubic Bezier curve interpolation [40] according to the occlusion length. The same curve interpolation is also used for marker trajectory resampling (from 30 fps to 100 fps) to obtain more samples for measuring gait associated data (and also for benchmarking with the 100fps gold standard VICON). We measure step and stride length, stance and swing phase based on the resampled trajectories of heel, ankle, knee and hip markers as explained next.



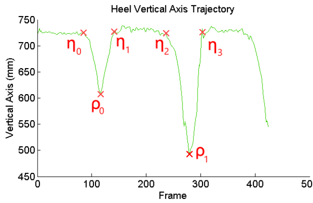


Fig. 6. Heel Horizontal Axis [25]

Fig. 7. Heel Vertical Axis [25]

1) Step and Stride Length: This task can be simplified into extracting stable values, where the heel marker trajectory

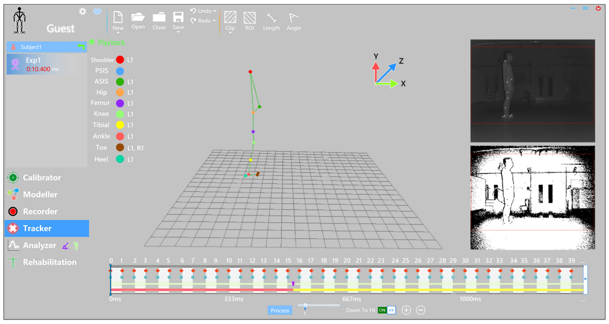


Fig. 8. Multi-view tracker snapshot.

horizontal axis value does not change over a window of frames (see an example in Fig.6, where ψ_0, ψ_1, ψ_2 denote three windows with no change detected using window matching between the inflection points). These points correspond to heel strikes to the floor. Once the left and right heel's horizontal stable values are found, the step and stride length can be calculated using the adjacent stable values over time, i.e., as $\psi_{i+1} - \psi_i$.

2) Gait Phases Detection: Gait events of heel strike and toe off are used to measure the stance and swing duration using heel marker trajectory vertical axis values. An example is shown in Fig.7, where η_0, η_1, η_2, η_3 denote inflection points, and ρ_0, ρ_1 local extremum, D_2 via our proposed global gradient filtering algorithm, Alg.6, instead of using an averaging filter as in [25]. The proposed algorithm quantizes the heel marker vertical axis trajectory and then searches each quantization region between the inflection points from the global minimum to the maximum by iteratively regrouping the scanned points. We heuristically set the quantization step-size (level resolution), $\gamma = 0.05$, the range left and right clip rate, $\xi = \phi = 0.33$, and the local range length boundary, $\tau = 3$ for extracting the inflection points and local peaks in order to obtain the relative time of heel strike and toe off through angle variation between the floor line and the line from toe to heel. A boolean variable 'locked' is used in Alg.6 as a flag for each range between two consecutive inflection points. For a given range p, in Step 18, $\min(p)$ and $\max(p)$'s lock levels denote the (local) minima (maxima) below (above) range p.

V. Visualization & Results

The proposed framework is demonstrated using an MS Kinect v2 sensor [11], though other sensors can be used, e.g., commercially available MS Kinect v1 [41], Intel RealSense R200 [13], SoftKinetic DepthSense Cameras [12]. The MS Kinect v2 sensor outputs 16-bit 512x424 pixel resolution of IR and depth images at 30fps. A user-friendly interface to the proposed underlying framework is designed for the proposed gait analysis application. It supports the following features: (1) Real-time camera/scene calibration. (2) Real-time subject modeling for frontal and sagittal plane. (3) Recording of IR and depth images using MS Kinect SDK v2.0 [11]. (4) Multi-view tracking and 3D trajectory reconstruction. (5) Kinematics analysis using customized function scripts. (6) Rehabilitation diagnostics interface using local or cloud database.

Algorithm 6: Inflection Points Searching

```
   Input: Vertical coordinate of heel marker trajectory $T$
            (e.g., see y-axis on Fig.7)
          Level resolution, $\gamma$;
          Range left clip rate, $\xi$;
          Range right clip rate, $\phi$;
          Local range length boundary, $\tau$;
   Output: Inflection points $\eta s$;
           Local peaks $\rho s$;
1  quantize $T$ using a step size $\gamma$ into quantization levels $\Omega$;
2  def range pool $P$ as an empty set;
3  set $\sigma = \min(T)$
4  for $k = \min(\Omega)$ to $\max(\Omega)$ do
5      find all ranges, i.e., differences between two values
         in $T$ that fall within the quantization bin $k$ and are
         smaller than $\sigma$; foreach found range $f = [f_l, f_r]$ do
6          set locked($f_l$) = false, locked($f_r$) = false;
7          if $f$ insides $P$ then
8              set $T = T \setminus f$ and goto 5;
9          if locked($f_l$) == true or locked($f_r$) == true then
10             update $P$ with $f$;
11         else
12             if $f_l$'s rate change $< \xi$ then
13                 set locked($f_l$) = true and goto 6;
14             if $f_r$'s rate change $< \phi$ then
15                 set locked($f_r$) = true and goto 6;
16     foreach $p$ in $P$ do
17         if (locked($p_l$) == true and locked($p_r$) == true) or
             $p$ is start or end range with one side locked then
18             if $length(p) > \tau$ and $\min(p\text{'s lock levels}) < 2 * \max(p\text{'s lock levels}) - p\text{'s initial local peak}$ then
19                 return $p$'s boundary values as inflection
                     points $\eta$ and initial local peak as
                     relevant local peak;
```

Fig.8 shows the snapshot of the software, which shows how convenient it is to access the recorded experiments by selecting the tracker tool. Users can also view the automatic reconstruction process within our multimedia application or manually playback the whole experiment. Autonomous analysis is performed and gait associated parameters are generated afterwards. These data (including joint angles, movement patterns, gait phases, step and stride length, swing and stance duration) can be easily accessed within the analyzer toolbox. For the rehabilitation application, a diagnostics interface is developed to report the patient's condition.

The proposed gait analysis application, and inherently the proposed framework and its six algorithms, was tested using 92 independent experiments with 14 subjects (11 males and 3 females), including 9 stroke survivors, and 25633 frames. Knee angle α, step length ζ, stride length ξ, stance and swing duration were measured as illustrated in Fig.9.

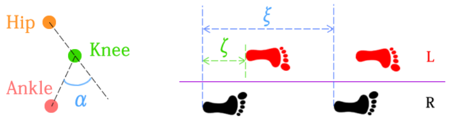


Fig. 9. Knee angle, step and stride length.

Evaluation of our proposed adaptive threshold analysis algorithm, Alg. 1, on an Intel i7-4710HQ 2.5GHz CPU, Windows10 OS, implemented by Visual C++, against the static-threshold marker detection algorithm of [25], [42] indicates higher detection accuracy as shown in Table II. However, it introduces an extra preprocessing step which increases the processing time by roughly 1ms per frame. In addition, the speed of Alg.1 depends on the physical distance between the marker and the sensor, which influences the size of the search region for the marker in IR image. This distance is dependent on subject body dimensions and their walking direction. However, as will be shown next, the proposed blob detection threshold analysis algorithm simplifies the following processing steps, making the overall processing faster.

TABLE II
PERFORMANCE OF THE PROPOSED DYNAMIC VS STATIC THRESHOLDING FOR MARKER DETECTION, SHOWING PERCENTAGE OF SUCCESSFULLY DETECTED MARKERS AND MEAN EXECUTION TIME PER FRAME.

Algorithms	Detected (%)	Time (ms/frame)
Static Threshold [25]	91.44 ± 3.52	0.15 ± 0.04
Adaptive Threshold	98.08 ± 1.08	1.21 ± 0.33

The performance of the proposed marker identification algorithm, Alg.2, is evaluated for each marker using recall rate of marker centroid's distance error (within error reference $\beta = 0.5, 1.5, 2.5$), that is, the average number of frames where the distance between the detected marker centroid and its true position is within β. Fig.10 shows the recall rate increments of 8 experiments for 12 markers using our proposed algorithm and the algorithm of [25] when $\beta = 0.5$ pixel. It can be seen that the recall rate has been improved by about 3∼9% especially for those markers that are attached to feet (ankle, toe, heel, another foot's toe and heel) where out-of-plane, motion blur are most likely to occur.

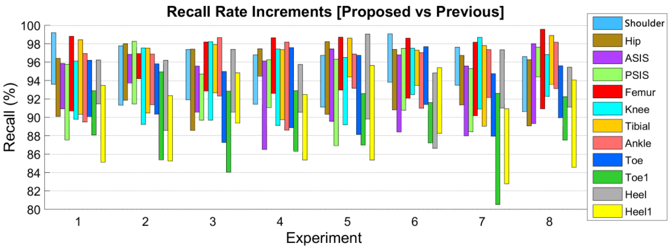


Fig. 10. Recall Rate Increments when $\beta = 0.5$ for 8 of experiments. Top boundary of each incremental rectangle is the recall rate using the proposed algorithm and bottom boundary is the one using the algorithm of [25].

Averaged results over all experiments and all markers are shown in Tab.III. It can be seen that the proposed Alg.2 significantly outperforms the previous approach in terms of both accuracy and the overall execution time.

TABLE III
PERFORMANCE OF THE PROPOSED ALGORITHM TO LOCATE CENTER OF A MARKER AVERAGED OVER ALL MARKERS AND 92 EXPERIMENTS.

Characteristics	Proposed	Previous [25]
Aver. callback (%), $\beta = 0.5$	95.86 ± 1.64	88.12 ± 2.61
Aver. callback (%), $\beta = 1.5$	97.32 ± 1.75	90.84 ± 2.67
Aver. callback (%), $\beta = 2.5$	98.04 ± 1.73	92.10 ± 2.45
Time (ms/frame)	7.72 ± 1.16	117.53 ± 17.94

Evaluation of accuracy of detecting gait events is performed by manually selecting the key frames and examining (with expert knowledge) the whole IR image sequence with corresponding static point clouds captured during the experiments, which are used as reference (i.e., ground truth), for validating the step and stride length, stance and swing duration. In order to evaluate the performance of swing phase detection, results were averaged to obtain the mean percentage error and percentage standard deviation in Table IV. It can be seen from the table, that the mean and standard deviation of the error are very small and slightly decreased with the proposed system compared to that of [25], attributed to the proposed Alg.6.

TABLE IV
PERFORMANCE OF THE TWO METHODS FOR MEASURING STEP AND STRIDE LENGTH, AND STANCE AND SWING PHASE

Metric	Step	Stride	Stance	Swing
Mean(%) [25]	1.06	1.16	1.81	1.12
Std(%) [25]	5.31	4.72	5.78	4.24
Mean(%)	0.98	1.08	1.73	1.09
Std(%)	4.12	4.23	4.56	3.56

The evaluation of the overall proposed system using VICON as a benchmark is discussed next. 5 subjects were simultaneously recorded using VICON and the proposed motion capture system. Each subject walked from left to right and back 8 times, hence a total of 40 experiments are used for comparison. The knee angle results for 4 experiments from 4 different subjects are shown in Fig.11. It can be seen that the angles from the proposed system are well within the 5 degree acceptable error margin compared to VICON.

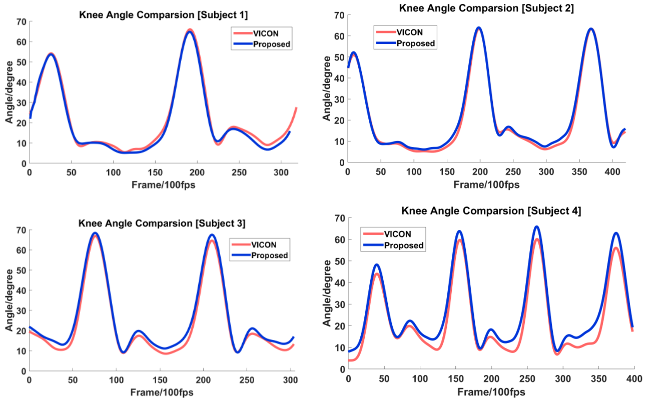


Fig. 11. Knee angle comparison with VICON.

The knee angle measurement performance is next evaluated by calculating root-mean-square error (RMSE) for each of the 40 experiments between the proposed framework and VICON. This is compared with the RMSE calculated from the system of [25] with VICON. The RMSE results are shown in Fig.12 as RMSE per experiment, where the effect of the algorithmic improvements over the previous system is clearly illustrated by reduced RMSE for all 40 experiments. Lower RMSE for the proposed system is attributed to the adaptive thresholding and the improved marker detection/labelling method. The maximum RMSE with the proposed system was under 6 degrees. Note that VICON returns joint trajectories instead of marker trajectories, thus a potential error comes from the misalignment between marker positions and actual joints.

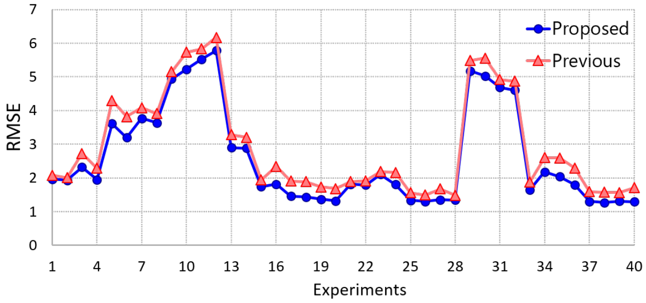


Fig. 12. Knee angle RMSE over 40 experiments.

Fig.13 shows two examples of knee joint angle during three walking cycles for two stroke survivors obtained by the proposed system. Comparing these results with those of the 4 healthy subjects shown in Fig.11, the effect of stroke is noticeable indicating movement abnormalities unique to each individual. This clearly shows the need for a person-centric framework, as proposed in this paper.

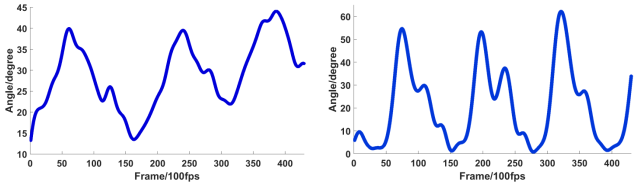


Fig. 13. Knee angle for two stroke survivors.

VI. Conclusions

A novel framework is proposed for motion assessment using a single depth camera based on simultaneous marker detection and identification in 3D space and model-based kinematics analysis. Both the optical motion capture system and gait analysis application are evaluated over close to 100 sequences, involving 9 stroke survivors and 5 healthy subjects, and benchmarked against the 12 camera state-of-the-art VICON system. In contrast to VICON and similar industrial standards, the proposed framework, which supports a portable sensor for capturing experiments, is suitable for tele-rehabilitation programs through our visualization, presentation and rehabilitation interfaces built in our proposed application. Validation results indicate high accuracy for sagittal plane gait analysis, which makes the system practical in clinical tests for different rehabilitation studies. Furthermore, our application-specific results clearly show the need for a person-centric framework, as proposed in this paper.

In the proposed framework, algorithms associated with optical motion capture are generic to any application while only Algs. 4, 6 and 5 are application specific, Hence, only the latter three need modification for different rehabilitation exercises that require motion analysis. While the results are presented for the rehabilitation walking exercise in the sagittal plane view only, the overall framework has also been tested for frontal view motion analysis (see for example [6] for assessing upper limb movement), where the same markers can be attached on both sides of the body and a frontal view model pre-configured similar to Fig. 5.

Future work comprises testing the framework performance using different depth cameras [13] and improving depth map recovery.

References

[1] P. Langhorne, F. Coupar, and A. Pollock, "Motor recovery after stroke: a systematic review," *The Lancet Neurology*, vol. 8, no. 8, pp. 741 – 754, 2009.

[2] A. Muro-de-la Herran, B. Garcia-Zapirain, and A. Mendez-Zorrilla, "Gait analysis methods: An overview of wearable and non-wearable systems, highlighting clinical applications," *Sensors*, vol. 14, no. 2, p. 3362, 2014.

[3] VICON, *Gait Analysis*, March 2016. [Online]. Available: http://www.vicon.com

[4] A. Leu, D. Ristic-Durrant, and A. Graser, "A robust markerless vision-based human gait analysis system," in *Applied Computational Intelligence and Informatics (SACI), 2011 6th IEEE International Symposium on*, May 2011, pp. 415–420.

[5] T.-Y. Liao, S.-G. Miaou, and Y.-R. Li, "A vision-based walking posture analysis system without markers," in *Signal Processing Systems (ICSPS), 2010 2nd International Conference on*, vol. 3, July 2010, pp. V3–254–V3–258.

[6] C. Yang, A. Kerr, V. Stankovic, L. Stankovic, P. Rowe, and S. Cheng, "Human upper limb motion analysis for post-stroke impairment assessment using video analytics," *IEEE Access*, vol. 4, pp. 650 – 659, 2016.

[7] C. Yang, U. Ugbolue, A. Kerr, V. Stankovic, L. Stankovic, B. Carse, K. Kaliarntas, and P. Rowe, "Autonomous gait event detection with portable single-camera gait kinematics analysis system," *Journal of Sensors*, vol. 2016, 2015.

[8] Y.-R. Li, S.-G. Miaou, C. Hung, and J. Sese, "A gait analysis system using two cameras with orthogonal view," in *Multimedia Technology (ICMT), 2011 International Conference on*, July 2011, pp. 2841–2844.

[9] J.-T. Zhang, A. C. Novak, B. Brouwer, and Q. Li, "Concurrent validation of xsens mvn measurement of lower limb joint angular kinematics," *Physiological Measurement*, vol. 34, no. 8, p. N63, 2013.

[10] T. G. Co, *M3D gait analysis system*, March 2016. [Online]. Available: http://www.tecgihan.co.jp/english/p7.htm

[11] *Kinect for Window software development kit*, March 2016. [Online]. Available: http://www.microsoft.com/en-us/kinectforwindowsdev

[12] *SoftKinetic DepthSense Camera*, March 2016. [Online]. Available: http://www.softkinetic.com/Products/DepthSenseCameras

[13] *Intel RealSense Camera (R200)*, March 2016. [Online]. Available: https://software.intel.com/en-us/RealSense/R200Camera

[14] S. Sarkar, L. Stankovic, A. Kerr, and P. Rowe, "Kinect-based lower limb motion analysis," in *ISB-2015 XXV Congress International Society of Biomechanics*, July 2015.

[15] S. kook Jun, X. Zhou, D. K. Ramsey, and V. N. Krovi, "A comparative study of human motion capture and analysis tools." [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.391.73

[16] H. Nguyen and J. Meunier, *Gait Analysis from Video: Camcorders vs. Kinect*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2014, vol. 8815, book section 8, pp. 66–73.

[17] R. Clark, S. Vernon, B. Mentiplay, K. Miller, J. McGinley, Y. Pua, K. Paterson, and K. Bower, "Instrumenting gait assessment using the kinect in people living with stroke: reliability and association with balance tests," *Journal of NeuroEngineering and Rehabilitation*, vol. 12, no. 1, p. 15, 2015.

[18] O. Lohmann, T. Luhmann, and A. Hein, "Skeleton timed up and go," in *Bioinformatics and Biomedicine (BIBM), 2012 IEEE International Conference on*, Oct 2012, pp. 1–5.

[19] E. Cippitelli, S. Gasparrini, S. Spinsante, and E. Gambi, "Kinect as a tool for gait analysis: Validation of a real-time joint extraction algorithm working in side view," *Sensors*, vol. 15, no. 1, p. 1417, 2015.

[20] M. Reyes, A. Claps, J. Ramrez, J. R. Revilla, and S. Escalera, "Automatic digital biometry analysis based on depth maps," *Computers in Industry*, vol. 64, no. 9, pp. 1316 – 1325, 2013.

[21] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys, "Accurate 3d pose estimation from a single depth image," in *Proceedings of the 2011 International Conference on Computer Vision*, ser. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 731–738. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2011.6126310

[22] M. Gabel, E. Renshaw, A. Schuster, and R. Gilad-Bachrach, "Full body gait analysis with kinect," in *Proceedings of EMBC 2012*. Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), August 2012. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=172555

[23] A. Leu, D. Ristic-Durrant, and A. Graser, "A robust markerless vision-based human gait analysis system," in *Applied Computational Intelligence and Informatics (SACI), 2011 6th IEEE International Symposium on*, May 2011, pp. 415–420.

[24] C. Kerrigan, M. Schaufele, and M. Wen, "Gait analysis," in *Rehabilitation Medicine: Principles and Practice*, 1998, pp. 167–174.

[25] M. Ye, C. Yang, V. Stankovic, L. Stankovic, and A. Kerr, "Kinematics analysis multimedia system for rehabilitation," in *New Trends in Image Analysis and Processing - ICIAP 2015*, vol. 9281. Springer International Publishing, 2015, pp. 1–9.

[26] L. Yang, L. Zhang, H. Dong, A. Alelaiwi, and A. El Saddik, "Evaluating and improving the depth accuracy of kinect for windows v2," *Sensors Journal, IEEE*, vol. 15, no. 8, pp. 4275–4285, Aug 2015.

[27] H. Samet and M. Tamminen, "Efficient component labeling of images of arbitrary dimension represented by linear bintrees," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, no. 4, pp. 579–586, Jul 1988.

[28] E. Lachat, H. Macher, M.-A. Mittet, T. Landes, and P. Grussenmeyer, "First experiences with kinect v2 sensor for close range 3d modelling," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-5/W4, pp. 93–100, 2015.

[29] F. Lenzen, H. Schfer, and C. Garbe, "Denoising time-of-flight data with adaptive total variation," in *Advances in Visual Computing*, ser. Lecture Notes in Computer Science, G. Bebis, R. Boyle, B. Parvin, D. Koracin, S. Wang, K. Kyungnam, B. Benes, K. Moreland, C. Borst, S. DiVerdi, C. Yi-Jen, and J. Ming, Eds. Springer Berlin Heidelberg, 2011, vol. 6938, pp. 337–346.

[30] W. Hu, X. Li, G. Cheung, and O. Au, "Depth map denoising using graph-based transform and group sparsity," in *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, Sept 2013, pp. 001–006.

[31] J. Shen and S.-C. Cheung, "Layer depth denoising and completion for structured-light rgb-d cameras," in *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013, pp. 1187–1194.

[32] Y. S. Kim, B. Kang, H. Lim, O. Choi, K. Lee, J. D. K. Kim, and C. Kim, "Parametric model-based noise reduction for tof depth sensors," pp. 82 900A–82 900A–8, 2012. [Online]. Available: http://dx.doi.org/10.1117/12.907614

[33] M. Camplani and L. Salgado, "Adaptive spatio-temporal filter for low-cost camera depth maps," in *2012 IEEE International Conference on Emerging Signal Processing Applications (ESPA)*, Jan 2012, pp. 33–36.

[34] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[35] C. Steger, "Subpixel-precise extraction of watersheds," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, 1999, pp. 884–890 vol.2.

[36] W. Förstner and E. Gülch, "A fast operator for detection and precise location of distinct points, corners and centres of circular features," in *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, 1987, pp. 281–305.

[37] U. Köthe, "Edge and junction detection with an improved structure tensor," in *Pattern Recognition*. Springer, 2003, pp. 25–32.

[38] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, pp. 79–116, 1998.

[39] S. Hinz, "Fast and subpixel precise blob detection and attribution," in *Proceedings of the 2005 International Conference on Image Processing, ICIP 2005, Genoa, Italy, September 11-14, 2005*, 2005, pp. 457–460. [Online]. Available: http://dx.doi.org/10.1109/ICIP.2005.1530427

[40] M. K. Agosto, "Bezier curves," in *Computer Graphics and Geometric Modelling: Implementation & Algorithms*, 2005, pp. 396–404.

[41] *Kinect for Xbox 360*, March 2016. [Online]. Available: http://www.xbox.com/en-US/xbox-360/accessories/kinect

[42] M. Ye, C. Yang, V. Stankovic, L. Stankovic, and A. Kerr, "Gait analysis using a single depth camera," in *Proc. IEEE GlobalSIP-2015, Orlando, FL*, Dec 2015, pp. 285–289.

Minxiang Ye received the B.Eng. degree in 2013, and is currently pursuing the Ph.D. degree in electronic and electrical engineering, both from the University of Strathclyde, Glasgow, UK. His research interests include image processing, machine learning and multimedia systems.

Cheng Yang (S11) received the B.Eng. degree in 2011, and is currently pursuing the Ph.D. degree in electronic and electrical engineering, both from the University of Strathclyde, Glasgow, UK. His research interests include multimedia processing and systems for healthcare.

Vladimir Stankovic (M03-SM10) received the Dr.- Ing. (Ph.D.) degree from the University of Leipzig, Leipzig, Germany in 2003. From 2003 to 2006, he was with Texas A&M University, College Station, first as Research Associate and then as a Research Assistant Professor. From 2006 to 2007 he was with Lancaster University. Since 2007, he has been with the Dept. Electronic and Electrical Engineering at University of Strathclyde, Glasgow, where he is currently a Reader. He has co-authored 4 book chapters and over 160 peer-reviewed research papers. He was an IET TPN Vision and Imaging Executive Team member, Associate Editor of IEEE Communications Letters, member of IEEE Communications Review Board, and Technical Program Committee co-chair of Eusipco- 2012. Currently, he is Associate Editor of IEEE Transactions on Image Processing, Editor at Large of IEEE Transactions on Communications, and Area Editor of Elsevier Signal Processing: Image Communication. His research interests include multimedia processing for health monitoring, user-experience driven image processing and communications and energy disaggregation.

Lina Stankovic [M04 - SM12], Lecturer at the University of Strathclyde, obtained her BEng Hons in Electronic Communications Engineering and PhD from Lancaster University in 1999 and 2003, respectively. From May 2002 to Sept 2007, she was at Lancaster University, first as a Research Associate, then as a Lecturer. She has also worked with BT Labs Martlesham Heath on digital video streaming, and Philips Research Eindhoven on signal acquisition and processing from 2D optical discs. She has published 4 book chapters and 107 peer-reviewed research articles and was area editor for the Int. Journ. Electronics and Comms (Elsevier) 2010-2015. Her main research areas lie in smart monitoring from sensor network platforms and meaningful information extraction that is user-centric, focused on signal and image processing of biomedical data for motion assessment and algorithmic analysis of energy data to understand how people use appliances in the home.

Andrew Kerr practised as a physiotherapist in the National Health Service (UK) for 10 years before undertaking postgraduate study at the University of Nottingham and Glasgow Caledonian University where he also worked as a lecturer. In 2010 he joined the Biomedical Engineering Department at Strathclyde University to conduct clinical research in the area of stroke rehabilitation and has been awarded several grants in this area. His primary motivation in research is the understanding of human movement through biomechanics. Specifically his interests lie in how individuals with impaired movement e.g. stroke, perform transitions between movement patterns.