In this paper, we propose a framework for rapidly estimating three-dimensional human pose from two camera views. It is based on an evolutionary algorithm. This system can be applied straightforwardly to inexpensive smart devices and used to evaluate multiple individuals’ calisthenics with two or more smart devices. On the other hand, in order to verify and evaluate input, Evolutionary Strategy Consensus was introduced to estimate fundamental matrix between views. The result shows that two dimensional position of human joints can be estimated into three dimensional pose even with errors.

Keywords: Three-dimensional pose estimation, Evolutionary algorithm, Evolutionary Strategy Consensus

1. Introduction

With the development of technology, human life expectancy has attained its maximum in recorded history. However, this has resulted in another scenario wherein the percentage of the elderly in the world population is increasing. Studies have shown that cognitive capability decreases with the normal aging of humans. This scenario causes significantly more accidents and cognition-associated diseases [1]. Meanwhile, physical exercise is considered to be one of the most effective solutions for this issue. The results of observational studies in [2] demonstrated a strong relationship between physical exercise and cognitive capability, particularly in the elderly. Elders who continue exercising would be less likely to develop cognitive diseases than individuals who do not.

Based on this, most physical exercises generally require the assistance of therapists during practice, and individuals who exercise at home have less capability for evaluating their performances accurately. This has resulted in the increasing requirement for practice monitoring systems, which enable the elderly to practice at home individually at their convenient time.

As the key component, the pose estimating system plays the most important role in these practice monitoring systems. In contrast to other fields, for a physical exercise system, therapists evaluate human states according to the movement of their arms or legs. Meanwhile, most of the referent studies focused on the evaluation of human posture with high accuracy, rather than the convenience and affordability for ordinary users. The accuracy of results is maximized using specific cameras such as depth cameras, or high-quality computation units for processing.

To facilitate this process and its implementation, in this paper, we propose a framework that applies an evolutionary algorithm to optimize three-dimensional (3D) human pose from the main view. It provides two-dimensional (2D) image coordinates of human joints. This system is lightweight and can be implemented on inexpensive smart devices, whereby it can be applied conveniently by ordinary individuals at home. To reduce the error caused by faulty identification of human joint, an additional image view for fundamental matrix estimation is introduced for correction. Compared with those in previous studies, the proposed method is lightweight and can be implemented in simple smart devices. In addition, calibration for acquiring camera parameters (which was generally required for most referent studies) is unnecessary. In order to verify and correct 2D coordinates position, we introduced multiple view for improving the estimation result.

The remainder of this paper is organized as follows: Section 2 introduces related works on human pose estimation. Section 3 explains our proposed framework for estimating 3D human poses. It also explains our strategy for solving the problem of errors caused while acquiring a 2D human pose. In Section 4, the experimental results are provided to demonstrate our method. The conclusion and future extensions are presented in the final section.

2. Related Work

Vision-based estimation of human pose has become one of the most popular research topics in the past decades [8][13]. The methods of pose estimation can largely be divided into two categories depending on whether prior knowledge is used: model-based and model-free [6].

In the early years, researchers focused on identifying 2D human poses because of hardware limitations. In [9], human pose estimation was formulated as a jigsaw puzzle problem in which the body-part tiles maximally cover the foreground region, match local image features, and satisfy body plan and color constraints. However, 2D human pose estimation is occasionally insufficient because of the loss of spatial information.

The improvement of devices (e.g., the development of
depth cameras) has enabled the capture of 3D coordinates. This has, in turn, enabled the estimation of 3D human pose estimation [10]. In [11], the authors proposed a method to rapidly and accurately predict the 3D positions of body joints from a depth image by utilizing random forest algorithms. This system runs at 200 frames per second on consumer hardware and shows high accuracy on both synthetic and real test sets.

Rather than acquiring 3D data directly, a few researchers estimated 3D human poses by utilizing multiple views. In [12], the authors estimated the 3D pose of multiple humans using multiple calibrated cameras. They first created a reduced state space by triangulating the corresponding body joints obtained from part detectors in pairs of camera views. Then, they introduced a novel 3D pictorial structure (3DPS) model for inferring 3D human body configurations from the reduced state space.

### 3. Human pose estimation

Notwithstanding the remarkable performance of the previous methods mentioned above, many problems exist, particularly constraints and the computation cost of estimation. Either specific devices (such as depth cameras) or more than three cameras are necessary. In this section, we explain our proposed method to address these constraints. It can be implemented in at most two smart devices. In the first step, the lengths of the upper and lower arms are initialized. This is captured by the camera with the main view. The fundamental matrix between the main camera view and sub camera view is estimated by the coordinates of the captured joints. It is used to correct the coordinates of the fault human joints. The process flow of the proposed method is illustrated in Fig. 1.

#### 3.1. Acquisition of 2D human joint position

The first step in our proposed method is to acquire 2D human skeletons captured by smart devices. Although there are many restrictions on 3D human pose identification, 2D human pose estimation using deep neural networks still achieves a high accuracy [3]. [4] proposed a robust and real-time six-degree-of-freedom re-localization convolutional neural network called PoseNet. It is also capable of detecting a human body’s joints in real-time [5] with inexpensive devices.

We set up two smart devices as shown in Fig. 2. Each joint in the 3D space \( P_i \) is projected onto two images \( p_{1i} \) and \( p_{2i} \), which refer to two image coordinates.

It should be noted that it is necessary to define the length of the human arm in the initial step of the system. In this step, users are instructed to adopt the T-pose to verify \( l_1 \) and \( l_2 \), which represent the lengths of the lower and upper arms, respectively. These two values are used in the following steps.

#### 3.2. Correction of human joint with fundamental matrix

Notwithstanding the performance of 2D skeleton joint estimation, information loss is caused by projection from 3D to 2D. Meanwhile, certain joints are not detectable because of the accuracy. This could result in a large prediction error. In this section, we explain our strategy to solve this problem. It involves the introduction of an additional view to improve the accuracy and reduce the instability simultaneously.

As shown in Fig. 2, we denote the projection points of the arbitrary 3D joint \( P \) as \( p_1 \) and \( p_2 \). Thus, the following equations are satisfied:

\[
\begin{align*}
    w_1 p_1 &= w_1 \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = A[R_1|t_1]P \\
    w_2 p_2 &= w_2 \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = A[R_2|t_2]P
\end{align*}
\]

where \( R \) and \( t \) represent the rotational matrix and transform vector, respectively, and \( A \) is the projection matrix.
The relationship between \( P \) and \( p_1, p_2 \) can be calculated as follows:

\[
p_1^T F p_2 = 0 \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad (3)
\]

where \( F \) is the so-called fundamental matrix. It contains nine elements.

Considering the number of parameters, at least eight corresponding points are required to estimate the previous formula [2]. Once the \( F \) between the two camera views is verified, the epipolar line in one view is calculated by the corresponding point in another view:

\[
l_1 = F^T p_2 \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad (4)
\]

\[
l_2 = F p_1^T \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad (5)
\]

According to the epipolar geometry, the corresponding point of \( p_t \) should lie on the epipolar line \( l_2 \), i.e.,

\[
||l_1, l_2||^2 = ||F^T p_2, F p_1^T||^2 = 0 \quad \ldots \quad \ldots \quad (6)
\]

To address the error, for a captured time-series data of human pose, suppose that the image position of the joint in the main view \( p_t \) at time \( t \), is set as

\[
p_1^t = \begin{cases} p_1^t & \text{if } d_1 \text{ or } d_2 < \tau \\ p_1^{t-1} + v_1^t & \text{otherwise} \end{cases} \quad \ldots \quad (7)
\]

Where \( d_1 \) and \( d_2 \) represent \( ||F^T p_2, F p_1^T||^2 \) and \( ||F^T p_2, F (p_1 + v_1^t)^T||^2 \) respectively. For point in another view \( \hat{p}_1^t \), it is updated as

\[
\hat{p}_2^t = \begin{cases} \hat{p}_2^t & \text{if } d_3 \text{ or } d_4 < \tau \\ \hat{p}_2^{t-1} + v_2^t & \text{otherwise} \end{cases} \quad \ldots \quad (8)
\]

where \( d_3 \) and \( d_4 \) represent \( ||F^T p_2, F p_1^T||^2 \) and \( ||F^T p_2, F (p_1 + v_1^t)^T||^2 \) respectively, and \( \tau \) is the threshold of the estimated Sampson distance, \( v_1^t \) and \( v_2^t \) are defined as:

\[
v_1^t = \hat{p}_1^t - \hat{p}_1^{t-2} \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad (9)
\]

\[
v_2^t = \hat{p}_2^t - \hat{p}_2^{t-2} \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad (10)
\]

### 3.3. Estimation of fundamental matrix

For two different views, 3D point large \( P = [X, Y, Z]^T \) project on these two views as \( p_1 = [u_1, v_1, 1]^T \), \( p_2 = [u_2, v_2, 1]^T \) and \( p \) prime, there is a 3 by 3 fundamental matrix \( F \) that satisfies equation like this:

\[
\begin{bmatrix} u_1 & v_1 & 1 \end{bmatrix} \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = 0 \quad (11)
\]

Since the scale of fundamental matrix \( F \) is unfixed, 8 point-pairs are needed in order to estimate \( F \). The relationship of feature points is clear. For example, in fig.3 point of left wrist in view one must corresponds to point of left wrist in view 2. However, since possibility of errors caused by fault detection or occlusion, it is necessary to filter these point pairs, which is called outlier in fig.3.

In order to reduce outlier and estimate fundamental matrix by inlier point pairs, we introduced Evolutionary Strategy Consensus(ESSAC). The process flow is explained in Fig. 4.

### 3.4. Modeling of upper limbs

To explain our method clearly, we describe the upper limbs as an example (rather than the entire skeleton) in the following section.

As stated previously that we suppose the length of arms \( l_1 \) and \( l_2 \) are known and fixed, any posture for a single arm can be represented as:

\[
g^* = \{ \theta_1^*, \theta_2^*, \theta_3^*, \theta_4^* \} \quad \ldots \quad \ldots \quad \ldots \quad (12)
\]

This is illustrated in Fig. 5. To verify the movement by previous rotational angles, it is necessary to calculate the position of the joints (elbow and wrist in this study). As a common structure form [14], we utilized the Denavit-Hartenberg (DH) representation for calculating the position of the elbow and wrist by modeling the forward kinematics of human upper limbs.

The (DH) parameters are named after Denavit and Hartenberg who introduced this representation in 1955 [?, ?]. It calculates the coordinate transformation frame-by-frame and makes a list of parameters (four parameters for each transformation):
In this study, we utilized the steady-state genetic algorithm (SSGA), which is a type of genetic algorithm (GA) for the predicting arm movement. For the GA, the $i_{th}$ candidate agent can be represented as:

$$g_i = (\theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i)$$  

According to the previous section, the positions of the elbow and wrist can be calculated according to the previous DH representations as:

$$[Q_{\text{elbow}}, Q_{\text{wrist}}] = DH(g^i)$$

where

$$Q_{\text{elbow}} = [q_1^{1 \cdot i}, q_2^{1 \cdot i}, q_3^{1 \cdot i}]^T$$

$$Q_{\text{wrist}} = [q_1^{2 \cdot i}, q_2^{2 \cdot i}, q_3^{2 \cdot i}]^T$$

represents the prediction of the 3D positions of the elbow and wrist, respectively.

Considering the complexity of the calculation and errors that can be omitted, we assume that the projection matrix is

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Therefore, the fitness value of $i_{th}$ agent can be evaluated by the fitness function:

$$f(g_i) = \sum_{m=1}^{2} w_m \cdot ||AP_m - AQ_m^i||^2$$

where $w_m$ represents the weight of the different joints. It should be noted that only two values are defined: $Q_{\text{elbow}}, Q_{\text{wrist}}$. It is apparent that when $f^i$ is approximately zero, the more possibility of this agent would be.

**Algorithm 1 Steady State Genetic Algorithm**

initialize population

while termination criteria is reached do

select best and worst agent

randomly select agent

worst agent crossover with probability

worst agent mutation with probability

fitness calculation

end while
Table 2. Rotational range of joint variables for left and right arms.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Left arm</th>
<th>Right arm</th>
</tr>
</thead>
<tbody>
<tr>
<td>θ₁</td>
<td>-π/2</td>
<td>π/2</td>
</tr>
<tr>
<td>θ₂</td>
<td>-π/2</td>
<td>π/2</td>
</tr>
<tr>
<td>θ₃</td>
<td>-π/2</td>
<td>π/2</td>
</tr>
<tr>
<td>θ₄</td>
<td>-π/2</td>
<td>π/2</td>
</tr>
</tbody>
</table>

Table 3. Comparison of performance between the cases with and without fundamental matrix correction, based on DTW score.

<table>
<thead>
<tr>
<th></th>
<th>zero noise</th>
<th>10% noise</th>
<th>20% noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>without fundamental matrix correction</td>
<td>21.12</td>
<td>41.70</td>
<td>188.08</td>
</tr>
<tr>
<td>with fundamental matrix correction</td>
<td>15.11</td>
<td>19.29</td>
<td>23.78</td>
</tr>
</tbody>
</table>

In contrast to the standard GA (wherein all the agents need to be updated), in the SSGA, only the worst candidate is replaced with a candidate solution generated by the crossover and mutation. The process is shown in Algorithm 1. It is an elitist crossover in which an individual is selected randomly, and a new individual is generated by combining genetic information of the selected individual and that of the best one. The worst individual is updated as follows:

\[
\theta_i^{\text{worst}} \leftarrow \theta_i^{\text{c}} + (\alpha \cdot \frac{\theta_i^{\text{c}} - \theta_i^{\text{best}}}{\theta_i^{\text{c}} - \theta_i^{\text{best}}} + \beta) \cdot N(0, 1) \tag{23}
\]

Where \(c\) represents the randomly selected candidate. \(N(0, 1)\) is a random value of the Gaussian distribution with a mean of zero and variance of one. Based on the calculation of SSGA, the best agent is selected to represent the current arm gesture after the iterations.

4. Experimental result

We first consider that the feasible moving range for human arms is limited. Furthermore, we set up the range of rotational angles for the two arms as shown in Table 2.

To test the performance with noise, we mixed noise of different amounts. To evaluate the similarity between two poses, we introduced dynamic time wrapping (DTW) as an evaluation algorithm. The DTW score tends to zero if the two poses are similar. A comparison of DTW scores is shown in Table 3. It is apparent that the two methods perform almost similarly when no noise is included. However, the performance would be higher because the noise is increased.

In the next step, we attempted to apply it to a real implementation. Considering the computational capability and cost, in this experiment, we utilized the iPod touch (7th generation) as the main device. PoseNet is capable of identifying human poses by detecting 17 key points in human joints. In this study, we tested only the exercises of the upper body. Therefore, only six joints were considered: left shoulder, left elbow, left wrist, right shoulder, right elbow, and right wrist. It is capable of performing the entire system, including the PoseNet deep neural network module with an FPS of 5–10.

In this experiment, we set the main iPod approximately 2 m in front of users and 1.5 m above the ground. The second iPod was parallel to the main iPod, which is rotated around the ground normal by approximately 15°. The corresponding joints are labeled and shown. Therefore, it is feasible to calculate the fundamental matrix directly.

We tested our proposed method with part of poses forms of calisthenics for rehabilitation in Japan. This exercise is used by elders to prevent the deterioration of bodily functions. Fig. 6 shows sample evaluated results. In 3D graphs, the blue and green lines represent the left and right arms, respectively. We also show the performance of two users. It is apparent that the poses of these two users have been identified with effective accuracy.

With noise included, we shows the performance in Fig. 7. The first row shows input with incorrect 2D position in two views, the second row shows the comparison between single view and multiple view. It is obviously that comparing with single view, multiple view shows the better result with incorrect input exit. However, we also shows the fault case in the last column. It is because incorrect input happened continuously that affect the prediction, therefore the multiple views did not work.

5. Conclusion

In this paper, we proposed a three dimension human pose estimation from two dimensional positions. The proposed method shows the good result, and the its low computation make it possible to run on inexpensive devices. By introducing multiple views, it is also possible to verify and correct incorrect input and improve the stability of the system.

Acknowledgements

This work was (partially) supported by JST [Moonshot R&D][Grant Number JPMJMS2034].

References:

Fig. 6. Performance of our proposed method for 3D pose estimation.

Fig. 7. Performance of our proposed method with incorrect inputs.