CIRP Annals - Manufacturing Technology 64 (2015) 9–12

Contents lists available at ScienceDirect

CIRP Annals - Manufacturing Technology

journal homepage: http://ees.elsevier.com/cirp/default.asp

Automated vision-based live ergonomics analysis in assembly operations

Jörg Krüger (2), The Duy Nguyen *

Institute for Machine Tools and Factory Management, Technische Universität Berlin, Pascalstr. 8-9, 10587 Berlin, Germany

ARTICLE INFO

Keywords:
Assembly
3D-Image processing
Ergonomics

ABSTRACT

Manual analysis and optimisation of ergonomic parameters can be tedious when process and worker's body size variance is high. Automating this process would reduce workload and enable developing assistance systems for worker support. This paper presents a system which computes the positions of the parts of the body from input depth images and assesses ergonomics scores. The method is based on Particle Swarm Optimisation (PSO). By parallel processing on graphics hardware (GPU), the system is able to provide ergonomic feedback within a few seconds.

1. Introduction

Musculo-skeletal disorders (MSDs) pose a serious threat for employees in assembly leading to temporal or chronic working inability. According to Parent-Thirion et al. [1], 22.8% of EU assembly workers suffer from muscular pain and 24.7% from lower back pain. One factor causing MSDs is awkward working posture. In order to be able to apply effective countermeasures, such as a re-design of the workplace or introducing robot assistance, a deep understanding of the underlying causes is essential.

Therefore, ergonomics assessment methods have been developed. These tools can be classified into screening methods (e.g. RULA [2]) and expert methods (e.g. EAWS [3]). Methods from the first group provide less insightful analyses, but pose less workload than the ones from the second group. Nevertheless, all methods still require high time effort due to tedious manual analysis of video material.

There have been approaches to develop computer aided assessment tools. Enomoto et al. [4] propose the assessment of ergonomics by simulating the process using a human model. Therefore, ergonomics parameters can be already estimated in planning phase. Ding and Hon [5] use a digital human model in order to simulate assembly sequences and choosing the ergonomically best ones out of it.

As the posture information of a real human subject has not been considered in these works they are suitable for assembly process planning rather than for assessing processes that are being performed at the moment. Measuring methods apply sensors to automatically obtain the information required for an analysis right in the process. CUELA [6] identifies the angle of the worker's back during the process and derives the ergonomic risk from this measurement. However, the system requires the subject to wear expensive and bulky equipment with including data cables attached to them.

To avoid the usage of additional movement limiting equipment, Ong and Wang [7] proposed to use vision-based tracking methods. In their work, the authors apply a technology to detect and classify bare hand interactions of the user with a virtual model of the product.

This paper presents a system which assesses posture related ergonomics by using the Microsoft Kinect® sensor, a low-priced consumer depth camera. The novelty is that the system can be used during the performance of a process and is able to provide feedback within a few seconds. The processing speed enables the usage of this system for the assistance of the worker such as proposed in Nguyen et al. [8]. Moreover, the method is able to identify the body parameters without markers attached giving the worker natural freedom in the movement.

This paper is structured as follows: Section 2 describes the technical details of the system. Section 3 provides experimental results and discusses them. Section 4 concludes the achievements of this work and provides an outlook for the future.

2. System description

The task of the system is to continuously output a numeric posture-based ergonomics criterion from a stream of depth images containing the subject at work. The steps to achieve these values are:

1. extracting the features from the depth image to identify the posture class;
2. classifying the posture; and
3. assigning an ergonomics score to the sequence of classified postures.

2.1. Feature extraction

The goal of this step is to obtain unique features which clearly distinguish different posture classes from each other but appear to be similar for instances of the same class. A feature vector representing the posture class is computed from an input depth image containing the segmented subject (e.g. by background subtraction). A classifier is trained using body angle features

* Corresponding author.
E-mail address: theduy.nguyen@iwf.tu-berlin.de (T.D. Nguyen).

http://dx.doi.org/10.1016/j.cirp.2015.04.046
0007-8506/© 2015 CIRP.

(see Fig. 1 left for an example). Body angles are computed between neighbouring body segments, e.g. upper leg and lower leg. An advantage of this type of feature is body size invariance. The body angles have the same value for the same posture independently from the size of the body's segments.

A robust position estimation of the joints is required to compute these angles. An approach inspired by Oikonomidis et al. [9] is used. The core idea is to find the parameters of a human model, which has created the observable input depth image. In other words, the model has to be determined, whose artificially created depth image is most similar to the depth image from the camera. The human model consists of a kinematic skeleton with the degrees of freedom of a human body. Ellipsoid wrappers located around each segment of the skeleton represent the "skin" (see Fig. 1). Although ellipsoid wrappers do not best describe the appearance of the human body, these geometric primitives have been chosen since creating depth images from ellipsoids represents an acceptable compromise between computational efficiency and similarity to the human anatomy.

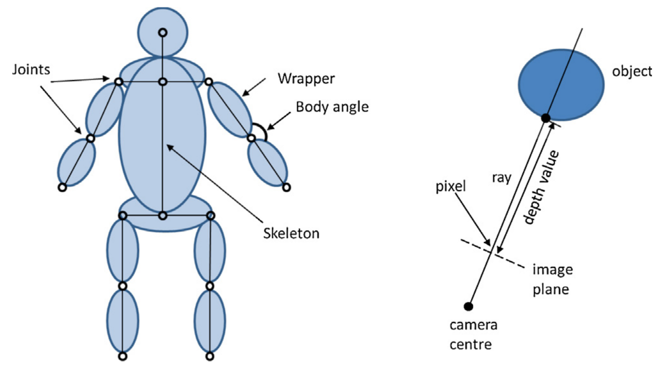

Fig. 1. Left: Schematic visualisation of the model with terminology used in this paper. Right: Basic principle of ray-casting technique in 2D.

An artificial depth image is created by ray-casting (see Fig. 1 right), a method from computer graphics which is able to create images with a high level of detail. Rays from the camera centre are traced through each pixel of the image plane. Their intersection point to each wrapper surface is computed. The depth value is the length of the vector from pixel to the point of intersection with the nearest wrapper.

For a better understanding of the algorithm, the formulas necessary to create the depth image of an ellipsoid are going to be derived. First, consider the intersection between a spherical wrapper and a ray with the origin $\mathbf{e}$ and the direction vector $\mathbf{d}$. The wrapper has a radius of 1 and its centre lies in the origin (0, 0, 0). Each point on the ray can be described as

$$\mathbf{e} + t\mathbf{d} \tag{1}$$

For each point p on the surface of the sphere, the following proposition applies

$$p * p = 1 \tag{2}$$

where '*' denotes a scalar product. Combining the Eqs. (1) and (2), results in the proposition for an intersection point is

$$\begin{aligned} (\mathbf{e} + t\mathbf{d})(\mathbf{e} + t\mathbf{d}) &= 1 \\ t^2(\mathbf{d} * \mathbf{d}) + t(2\mathbf{e} * \mathbf{d}) + (\mathbf{e} * \mathbf{e} - 1) &= 0 \end{aligned} \tag{3}$$

This equation can be solved using the quadratic formula

$$t* = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \tag{4}$$

with $a = \mathbf{d}^*\mathbf{d}$, $b = 2\mathbf{e}^*\mathbf{d}$ and $c = \mathbf{e}^*\mathbf{e} - 1$. If there is no intersection point between sphere and ray, the term under the square root will become negative.

In order to compute the intersection point of the ray with an arbitrary ellipsoid, the original Eq. (3) has to be modified. Given an ellipsoid with the centre $\mathbf{c} = (C_X, C_Y, C_Z)$, the rotation matrix $\mathbf{R}$ and the diagonal scaling matrix $\mathbf{S} = diag(S_X, S_Y, S_Z)$ describing the extent of the ellipsoid in the three directions, the ray is first transformed by

$$\begin{aligned} \mathbf{e}* &= \mathbf{S}^{-1}\mathbf{R}^{-1}(\mathbf{e} - \mathbf{c}) \\ \mathbf{d}* &= \mathbf{S}^{-1}\mathbf{R}^{-1}\mathbf{d} \end{aligned} \tag{5}$$

The intersection point is then computed by using Eq. (3) with the transformed ray parameters $\mathbf{e}^*$ and $\mathbf{d}^*$.

This procedure has to be accomplished for each pixel and wrapper. The number of ray-casting operations is the number of pixels times the number of wrappers. The final depth value for one pixel is the one computed from the smallest positive t of intersections of all wrappers. Since depth image creation is computationally demanding, the method is sped up by parallel processing on GPU. Given n pixels or rays with the parameters $\mathbf{d_1},\ldots,\mathbf{d_n}$ and $\mathbf{e_1},\ldots,\mathbf{e_n}$, multiple ray-surface intersections can be independently computed in one step. Afterwards, the minimum depth value among each wrapper is search in parallel for each pixel. With this technique, feedback times have been reduced from several minutes on a CPU to a few seconds per image on a recent GPU.

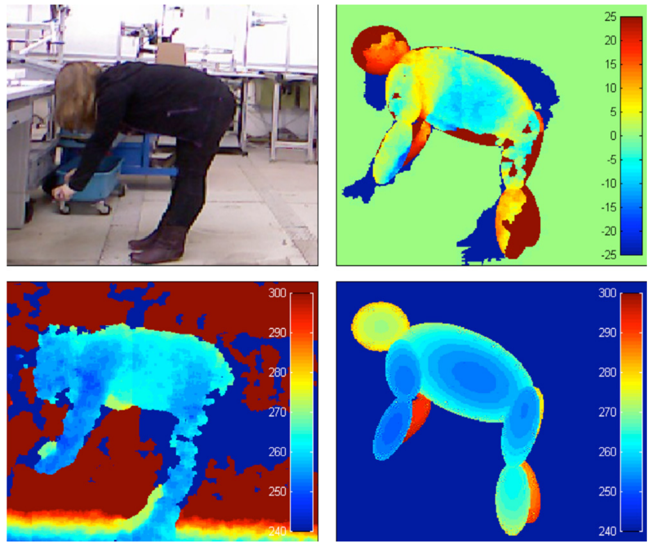

Fig. 2. Top: RGB image (left) and difference image between raycasted depth image and Kinect depth image (right). Bottom: Kinect depth image (left) and raycasted depth image (right). The colours in the depth images encode the depth value in cm.

Having created the artificial depth image I_{Raycast} (see Fig. 2 bottom left for example), it is compared to the original one I. The distance measure between the images is a weighted sum of three terms

$$D(I, I_{\text{Raycast}}) = w_{\text{Depth}}D_{\text{Depth}} + w_{\text{Overlap}}D_{\text{Overlap}} + w_{\text{Feet}}D_{\text{Feet}} \tag{6}$$

D_{Depth} is the mean difference between the depth values

$$\begin{aligned} D_{\text{Depth}} &= \sum_{x,y} \delta(x,y)\left|I(x,y) - I_{\text{Raycast}}(x,y)\right| \\ \delta(x,y) &= \begin{cases} 1, I(x,y) > 0 \wedge I_{\text{Raycast}}(x,y) > 0 \\ 0, \text{else} \end{cases} \end{aligned} \tag{7}$$

D_{Overlap} is the number of pixels which have only a value in one of the images

$$\begin{aligned} D_{\text{Overlap}} &= \sum_{x,y} \phi(x,y) \\ \phi(x,y) &= \begin{cases} 1, I(x,y) = 0 \wedge I_{\text{Raycast}}(x,y) > 0 \\ 1, I(x,y) > 0 \wedge I_{\text{Raycast}}(x,y) = 0 \\ 0, \text{else} \end{cases} \end{aligned} \tag{8}$$

The first two terms are applied from Ref. [9]. In order to adapt the original hand tracking algorithm to full body tracking, the third term has been added. D_{Feet} is the difference in height (y-value) between the feet of the model and the lowest points of the subject mask in the input image to avoid solutions where the model's feet do not touch the ground.