engineering proceedings

MDPI

Proceeding Paper

Using an Ensemble of Deep Neural Networks to Detect Human Keypoints in the Workspace of a Collaborative Robotic System †

Yuri Ivanov *, Sergey Zhiganov, Mikhail Gorkavyy, Sergey Sukhorukov and Daniil Grabar

Department of Energy and Management, Komsomolsk-na-Amure State University, 681013 Komsomolsk-na-Amur, Russia

* Correspondence: ivanov_ys@icloud.com; Tel.: +7-4217-241-139; Fax: +7-4217-536-150

† Presented at the 15th International Conference "Intelligent Systems" (INTELS'22), Moscow, Russia, 14–16 December 2022.

Abstract: It is suggested that the use an ensemble of deep neural networks can determine the spatial position of the operator using keypoints with a multicamera sensor system. The advantage of the algorithm is the use of a multicamera system that allows keypoints to be linked to the local coordinate system of an industrial robotic complex. The testing of this work was made on the basis of modern embedded computing hardware and software. The effectiveness of the proposed approach is demonstrated even when only a subset of key points is found in the frame, as well as when they partially overlap. A software module in Python has been developed for detecting and localizing key points of the operator and industrial manipulator. The proposed approach will make it possible to plan the robot's trajectories for the safe execution of joint operations in one workspace. The developed algorithm will be used to predict the operator's actions in the workspace and detect abnormal situations and possible intersections in the trajectories of the collaborative robot.

Keywords: detection; recognition; classification; human pose estimation; deep neural network; video stream; keypoint detection

check for updates

Citation: Ivanov, Y.; Zhiganov, S.; Gorkavyy, M.; Sukhorukov, S.; Grabar, D. Using an Ensemble of Deep Neural Networks to Detect Human Keypoints in the Workspace of a Collaborative Robotic System. *Eng. Proc.* **2023**, *33*, 19. https://doi.org/10.3390/engproc2023033019

Academic Editors: Askhat Diveev, Ivan Zelinka, Arutun Avetisyan and Alexander Ilin

Published: 13 June 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The human pose estimation task is one of the most interesting areas of research. It has applications in various fields, such as games, healthcare, augmented reality and sports [1]. The operator position estimation is of great importance in collaborative robotics, since the solution of this task will increase the efficiency of robotics and expand the possibilities of its application. A real-time response is required not only in security applications [2,3], where human actions must be detected in time, but also in industrial applications where human movement is predicted, to prevent collisions with robots in shared workspaces. For these reasons, research on motion-capture systems without markers and wearable sensors has been especially in demand in recent years.

When solving the problem using computer vision methods, it is necessary to find the coordinates of each joint (arm, head, torso, etc.), called keypoints, in the video frame, and form a skeletal representation of the human body. Thus, the detection of keypoints includes the simultaneous detection of people and the localization of their keypoints.

This problem is particularly difficult due to the heterogeneity of objects that have various and potentially complex shapes, as well as the difficulties arising from background noise and partial overlaps between objects (occlusions).

Thus, the most crucial problems can be pointed out, which are [4]:

- variability of lighting conditions;
- partial occlusions and layering of objects on the scene video;
- the complexity of the human skeleton structure;

- loss of three-dimensional information that occurs as a result of observation from one point, etc.

There are several approaches to the human body modeling:

- model based on the skeleton;
- contour model;
- three-dimensional model.

Though the posture estimation has been studied for many years, this task still remains very complex and largely unsolved. There is still no universal approach that can give satisfactory results in general non-laboratory conditions.

The paper [5] considers a linear Tensor-on-Tensor regression model to predict human behavior. However, in this work, only reference points and connections of the human upper limbs are constructed, i.e. only hand movements in the working area are analyzed.

An approach to detect hazardous operator behavior was proposed in [6]. The authors simulate dangerous behavior through time series analysis to detect hazards. Unfortunately, the authors did not give examples of real testing of the proposed algorithms, and all tests were carried out only in simulation.

Industrial cobots must be able to detect the human presence, identify and determine intentions based on hand gestures, actions, etc. There is a promising tendency here towards marker-less recognition [7]. High-level motion planning is usually combined with the ability of a robot to recognize the intentions of a human partner [8–10].

In [11] the authors applied the promising technology ViT (Visual Transformers) and obtained the maximum score in the Pose Estimation on MPII Human Pose benchmark [12].

The approaches that use multiple sensors or camera networks appear to be promising, which increase reliability in the case of occlusion and visual field limitations.

The paper [13] proposed an approach to estimate 3D poses of multiple persons in calibrated RGB-Depth camera networks. Each single-view outcome was computed by using a CNN for 2D pose estimation and extending the resulting skeletons to 3D by means of the sensor depth. The authors presented their solution in the form of an open-source library OpenPTrack [14].

The study [15] proposed a convolutional neural network (CNN) approach for estimating human body pose using a small number of cameras, including outdoors scenes. The authors [16] proposed a purely geometric approach to infer a multiview pose from a synchronous set of 2D skeletons.

The paper [17] proposed an approach to estimate a 3D human pose from multi-view video recordings, taken with unsynchronized and uncalibrated cameras. A unique approach to self-calibrate the system using the detected keypoints was employed in the research.

It should be noted that despite the high accuracy of some approaches, they are rather resource intensive and involve the transmission of a video stream and calculations using server graphics processing units (GPUs). However, in industrial systems, the necessity to use embedded computing modules based on GPUs of lower performance as on-board computers for robots arises. The fact imposes significant restrictions on the complexity of algorithms due to their low power consumption and small size.

It is important to emphasize once again that the application possibilities of cobots will be significantly increased thanks to timely recognition, analysis and prediction of human actions based on data from a multimodal sensory system and the development of control actions, taking into account emergency situations and extreme conditions in real time. Thus, the technological solutions, including the development of software and algorithmic support for the implementation of joint work of various types of robots with a person, is a promising task.

From our point of view, the most promising is the approach that combines a multi-camera system, as well as additional sensor devices, thereby providing multimodal data processing based on neural network approaches. For this reason, this article proposes to

use an ensemble of deep neural networks (NN) to determine the spatial operator posed using keypoints and to link them to the world coordinate system of the robot.

The scientific novelty of the project is in the proposed set of methods, approaches and algorithms aimed at ensuring effective interaction between the components of the operator-cobot system.

2. Problem Statement

The task of increasing the efficiency of cobot-person interactions is formulated as follows. Judging by the incoming video stream from surveillance cameras, microphones, and other sensors installed in the working area, it is necessary to recognize, localize in space and build a predictive dynamic model of the operator's behavior, which should be further synchronized and adapted to the collaborative control system of the manipulation robot with a different number of degrees of freedom to perform joint, diverse and previously unknown scenarios.

The research deals with the problem of detecting and localizing human keypoints. Thus, the particular task solved in this article can be formulated as follows.

By the incoming video stream from several surveillance cameras, it is necessary to recognize and localize keypoints of the operator in three-dimensional space. To preliminarily calibrate the system, it is necessary to apply calibration methods "by template" and to apply computer vision methods based on deep neural networks in order to localize keypoints.

One needs to develop software in the Python environment and conduct a full-scale experiment with video scenes which come from surveillance cameras installed at a production facility.

Mathematical Formulation of the Keypoint Detection Task

The task of detecting human keypoints in an image can be formulated as a regression task.

Let there be: a set of images $\omega \in \Omega$, defined using features x_i, $i = 1, n$, the totality of which for the image ω is represented by vector descriptions $\mathbf{\Phi}(\omega) = (x_1(\omega), \dots x_n(\omega)) = \mathbf{x}$ and a set of values of the dependent variable $\mathbf{y}$, corresponding to them, each of which is a vector of values $\mathbf{y} = (y_1, y_2, \dots y_n)$, and $\mathbf{y}_i \in \mathbb{R}$.

A priori information is represented with a training set (dataset) $\mathbb{D} = ((\mathbf{x}^j, \mathbf{y}^j)), j = \overline{1, L}$, defined using a table, each row j, contains a vector image description $\mathbf{\Psi}(\omega)$ and the value of the target variable. Note that the training set characterizes the unknown mapping ${}^*\mathbf{F} : \Omega \to \mathbb{Y}$.

We specify the regression task to the keypoints detection task. Let there be a video stream frame $\mathbf{I}^t$, where t - is the number of the current frame. Judging by the available frames $\mathbf{I}^t$ of a continuous video stream $\mathbf{V} = (\mathbf{I}^1 \dots, \mathbf{I}^t, \dots, \mathbf{I}^\tau)$ and a priori information given by the training set $\mathbb{D} = ((\mathbf{x}^j, \mathbf{y}^j)), j = \overline{1, L}$ for deep learning of a supervised NN, it is required to solve the problem of image recognition: To detect the key points of an object on a video stream frame. Key points can be represented as a vector $\mathbf{y} = (y_1, y_2, \dots y_n)$, containing object coordinates.

Generally accepted metrics are used to evaluate the task of detecting the keypoints of a human as a performance criterion [18].

Average precision (AP), which is averaged over various Object Keypoint Similarity (OKS) thresholds is set to $0,50 : 0,05 : 0,95$.

OKS is calculated based on the distance between predicted points and ground truth points normalized to the human scale. The scale and constant of the keypoint is needed to equalize the importance of each keypoint: the location of the neck position is more precise than the position of the hips [19]: OKS $= exp(-\frac{d_i^2}{2s^2k_i^2})$, where d_i - Euclidean distance between true keypoint and predicted keypoint; s–scale. The square root of the object segment area; k–is a constant for the keypoint that controls the decline; each keypoint has its own coefficient (the circles for the shoulders and knees can be larger than for the nose or eyes). The OKS metrics only shows how close the predicted keypoint is to the true keypoint (value between 0 and 1). Perfect predictions will have OKS $= 1$, while the predictions for which all keypoints differ by more than a few standard deviations $s \cdot k_i$, will have

OKS ≈ 0. Mean Average precision is also calculated using OKS with thresholds of 0.50 and 0.75 (AP50 and AP75).

Percentage of Detected Joints–PDJ. A detected joint is considered correct if the distance between a predicted keypoint and a true one is within a certain fraction of the diagonal of the bounding box (diameter of the torso).

The use of the PDJ metrics implies that the accuracy of all joints is estimated using the same error threshold.

$$\text{PDJ} = \frac{\sum_{i=1}^{n} bool(d_i < 0.05) * diagonal}{n},$$

where d_i–is the Euclidean distance between the true keypoint and the predicted keypoint; $bool(condition)$–a function that returns 1 if the condition is true, 0, if it is false; n–the number of keypoints in the image.

Percentage of Correct Key-points PCK [20] considers a body part to be correctly located if the estimated endpoints of the body segments are within 50% of the segment length of their true location. PCKh-0.5 [21]. PCK modification using 50% head segment length matching threshold.

3. Problem Solution

The algorithm which is aimed to detect and localize keypoints in the world coordinate system is divided into a separate subtasks solution (Figure 1):

1. Preliminary calibration of the multi-camera system and its binding to the cobot coordinate system is performed.
2. Human keypoints are detected on each of the cameras.
3. Aggregation of detected keypoints is performed and their coordinates in the world system are compared.

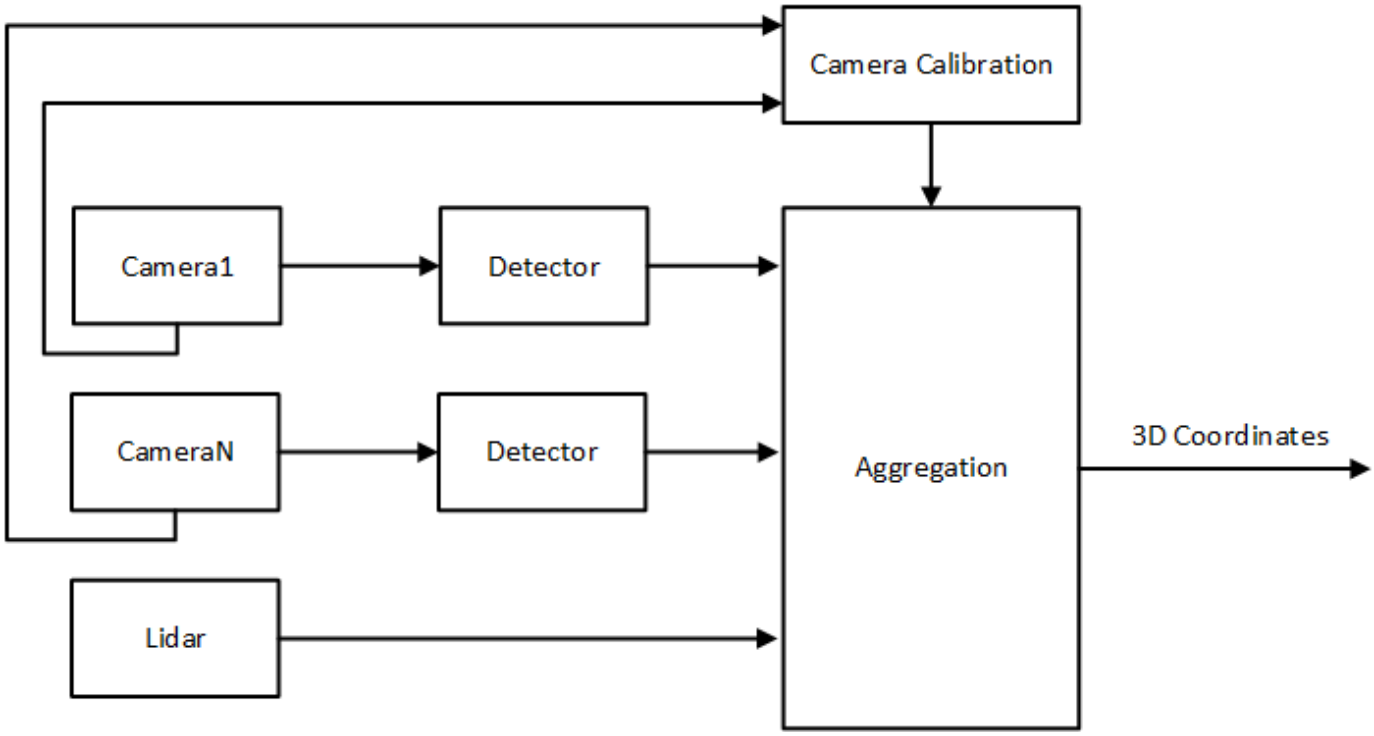
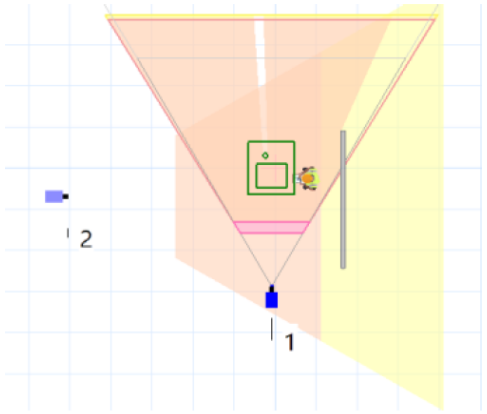


Figure 1. Algorithm for detecting and localizing keypoints in the world coordinate system

3.1. Calibration of the Multi-Camera System

Let there be a multi-camera system consisting of two cameras observing one scene. The configuration and location of the system is shown in Figure 2. The camera registers a

scene containing N reference points. The task is to use the three-dimensional coordinates of the reference points (X_i^p, Y_i^p, Z_i^p) and the coordinates of their projection in the camera image plane (c_{x_i}, c_{y_i}), where $i = 1...N$ evaluation of matrix elements **A**. As a rule, a calibration object in the form of a checkerboard is formed for camera calibration, since the use of alternating black and white squares has a sharp gradient in two directions. The intersections of the checkerboard lines are used as corners.





Camera 1: installation height – 1910 mm.,
distance to the table – 4700 mm.,
focal length - 4.7 mm.,

Camera 2: installation height – 1510 mm.,
distance to the table – 2210 mm.,
focal length – 1.5 mm.

Figure 2. Observed scene

K images for each camera are generated with a calibration object depending on the number of rotation angles N. To calculate four internal parameters $(f/w, f/h, c_x, c_y)$ where (c_x, c_y) are the natural coordinates of the point, f is the distance from the optical center, w, h are the dimensions along the axes ox, oy and six external parameters such as (ψ, ϕ, θ) rotation angles and (T_x, T_y, T_z) transfer parameters. The number of frames and the number of corners is calculated using the following expression: $2 \cdot N \cdot K \geq 6 + 4$. To obtain the coordinates of reference points (X_i^p, Y_i^p, Z_i^p) and the coordinates of their projection in the camera image plane (c_x, c_y) the *findChessboardCorners* function for finding corners on the chessboard is used, provided in the OpenCV library (Figure 3). Using the method [22] the matrix element of camera internal parameters is evaluated:

$$\mathbf{A} = \begin{bmatrix} f/w & 0 & c_{x_0} \\ 0 & f/h & c_{y_0} \\ 0 & 0 & 1 \end{bmatrix}.$$

The result of the function is a matrix of the camera internal characteristics **A**, distortion vector **r**, rotation vector **m** and transfer vectors **q**.

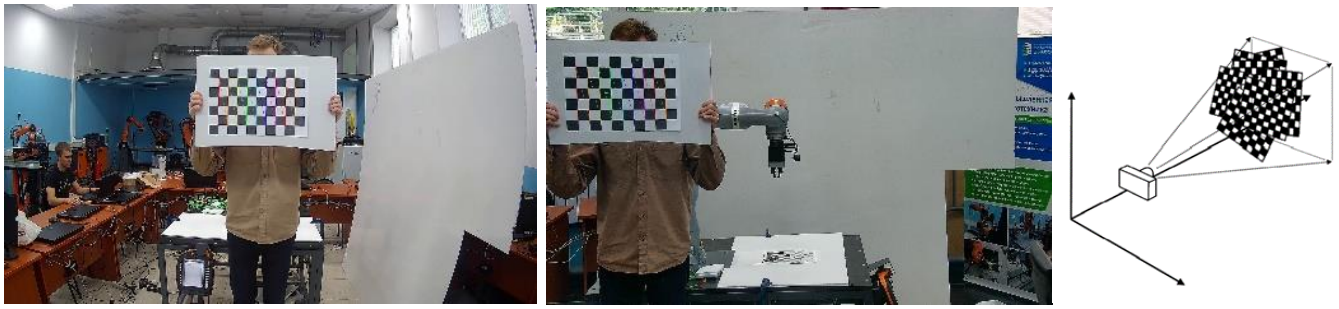


Figure 3. Camera calibration

3.2. Human KeyPoint Detection

The following approach worked out by the authors [23] based on the OpenPifPaf project [24] was used as an algorithm to detect the operator keypoints. The authors [23] proposed a number of modifications that improve the quality of keypoints detection. The

underlying OpenPifPaf is a bottom-up detector based on composite fields. The architecture of the model is shown in Figure 4. The input gets an $h \times w$ image with three color channels. The neural network encoder generates PIF and PAF fields 17×5 and 19×7 channels. The decoder converts the PIF and PAF fields into pose estimates containing 17 joints each. Each connection is represented by x and y coordinates and a confidence score. ResNet is used as encoders [25] or ShuffleNetV2 [26].

The Part Intensity Field (PIF) and Part Association Field (PAF) blocks are 1×1 convolutions followed by subpixel convolutions. These blocks are trained to detect and link key points. For architecture training, the COCO dataset was used, which can also be supplemented with synthetic data to adapt to a specific task. The method of the network retraining using only synthetic data with pretrained weights seems more promising.

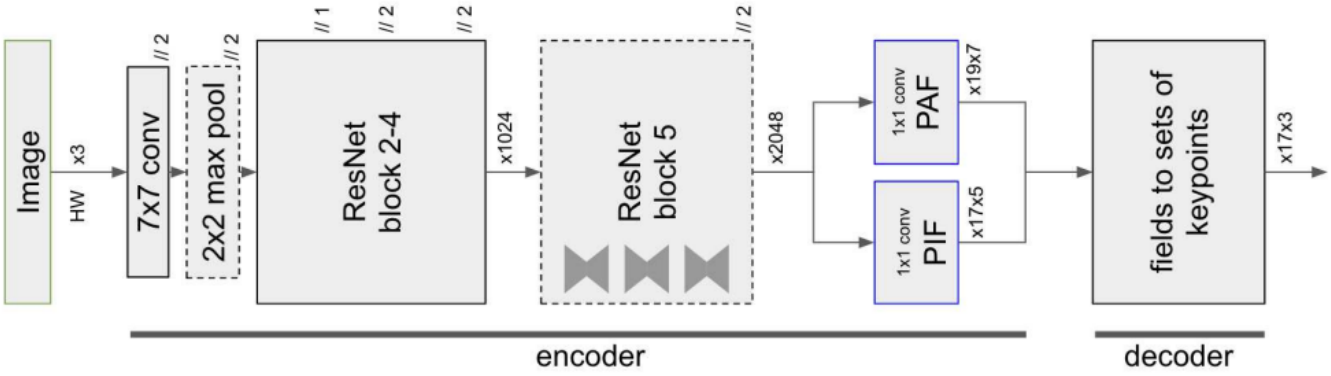


Figure 4. Keypoint detector architecture.

To obtain synthetic data, the Unity3D engine was used, as well as images obtained from the cameras of the observed scene. This approach allows you to adapt the algorithm to certain conditions.

Figure 5 shows the results of recognition on video surveillance cameras installed on the experimental site (a), as well as on synthetic data (b). As a result, the algorithm detected key points in the COCO format.

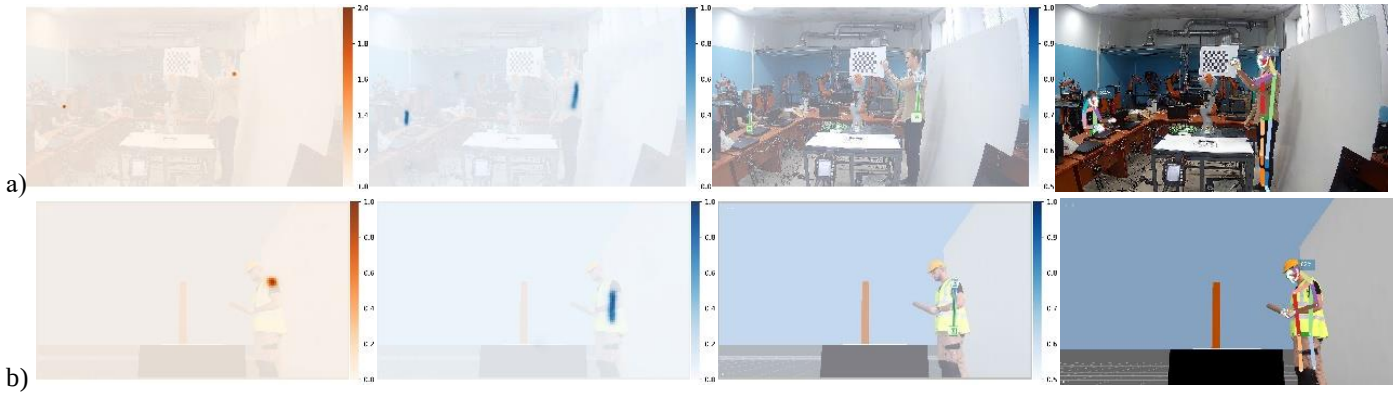


Figure 5. Keypoint detector: (**a**) real camera (**b**) synthetic data.

3.3. Mapping Keypoints in 3D Space

The result of the second stage was the keypoints of the operator detected on the images of each of the surveillance cameras. It was necessary to map 2D coordinates to the points in 3D space of the world coordinate system.

There are methods to calculate the 3D position of points for stereoscopic systems consisting of two cameras, the optical axes of which are parallel, and the straight line which passes through the optical centers is perpendicular to the optical axes. In this case, to obtain

a point in three-dimensional coordinates, it is necessary to calculate the reference points using the following expression:

$$x_1^1 = \frac{f(X^w + \frac{b}{2})}{Z^2}, \quad x_2^2 = \frac{f(X^w - \frac{b}{2})}{Z^2}, \quad y_1^1 = y_2^2 = \frac{fY^w}{Z^w},$$

where X^w, Y^w, Z^w are world coordinates of the point, x_1, y_1 are the projection coordinates in the image plane of the first camera, while x_2, y_2 for the second camera. Then the point coordinates in three-dimensional space will be as follows:

$$X^{w^c} = b\frac{(x_1^1 + x_2^2)}{2(x_1^1 - x_2^2)}, \quad Y^{w^c} = b\frac{(y_1^1 + y_2^2)}{2(x_1^1 - x_2^2)}, \quad Z^{w^c} = \frac{fb}{x_1^1 - x_2^2},$$

where f is the distance from the optical center, b is the length of the straight line segment between the optical centres. In the case when the camera axes are not parallel and the direction of cameras optical center displacement is arbitrary, the calculation of the point coordinates for any camera presupposes that the following parameters should be calculated:

$$\mathbf{v}_1 = \frac{\mathbf{A_1 M_1}}{Z_1}, \quad \mathbf{v}_2 = \frac{\mathbf{A_2 M_2}}{Z_2},$$

$$\begin{bmatrix} Z_1^t \\ Z_2^t \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1^T \mathbf{A}_1^{-T} \mathbf{A}_1^{-1} \mathbf{v}_1 & -\mathbf{v}_1^T \mathbf{A}_1^{-T} \mathbf{R}^T \mathbf{A}_2^{-1} \mathbf{v}_2 \\ -\mathbf{v}_1^T \mathbf{A}_1^{-T} \mathbf{R}^T \mathbf{A}_2^{-1} \mathbf{v}_2 & \mathbf{v}_1^T \mathbf{A}_1^{-T} \mathbf{A}_1^{-1} \mathbf{v}_1 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{v}_1^T \mathbf{A}_1^{-T} \mathbf{R}^T \\ \mathbf{v}_2^T \mathbf{A}_2^{-T} \end{bmatrix} \mathbf{t},$$

where $\mathbf{M}_1, \mathbf{M}_2$ characterize the coordinates of a certain point in three-dimensional space in the coordinate system in the system of the first and second cameras, $\mathbf{A}_1, \mathbf{A}_2$ are the matrix of internal parameters of the first and second cameras. $\mathbf{R}$ is an orthogonal matrix that describes the orientation of the coordinate system of the second camera relative to the first one, $\mathbf{t}$ is the translation vector that determines the position of the optical center of the second camera in the coordinate system of the first one. The obtained parameters can be used to get a vector of 3D point coordinates for any of the cameras: $\mathbf{M}_1^p = Z_1^t \mathbf{A}_1^{-1} \mathbf{v}_1$, $\mathbf{M}_2^p = Z_2^t \mathbf{A}_2^{-1} \mathbf{v}_2$.

4. Semi-Natural Experiment

The proposed approach was implemented in Python using the pytorch library. The following computer configuration was used for testing: Intel Core i5, 8 Gb RAM, Nvidia Geforce 1080 Ti. Figure 6 demonstrates the algorithm in action in the current industrial line of the robotics center.

It has been experimentally proven that this approach allows the pose to be restored when only a subset of keypoints are in the video scene, as well as when they partially overlap.

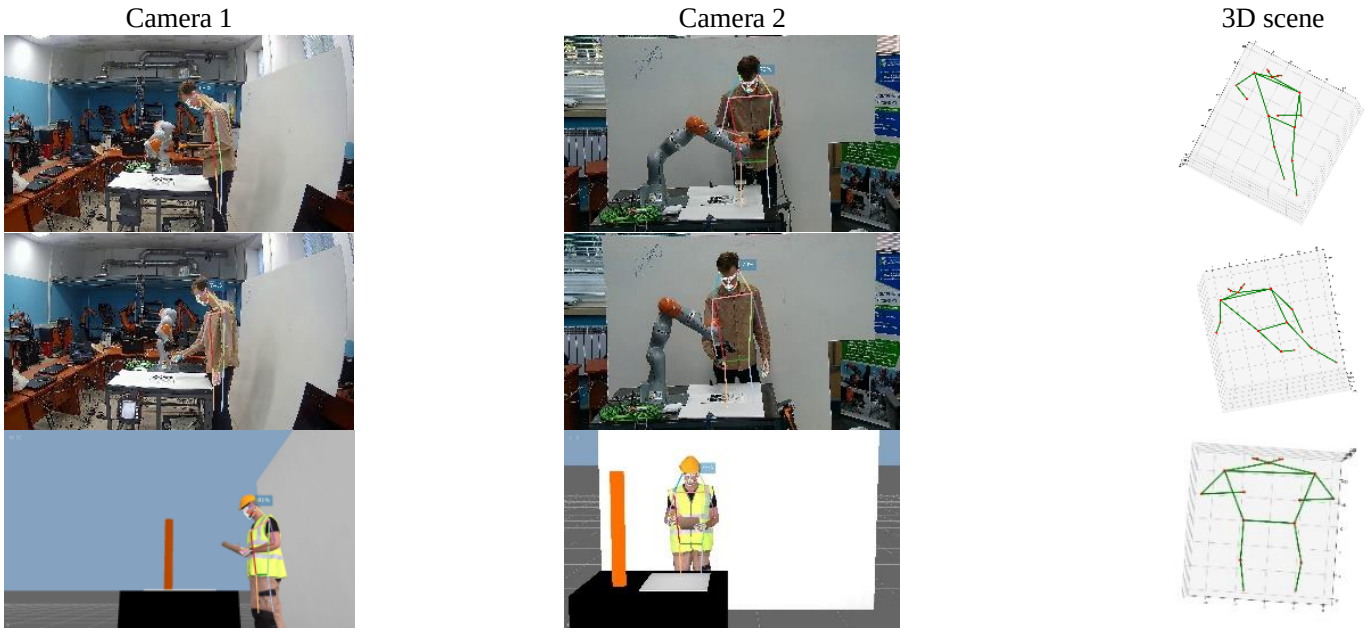


Figure 6. The proposed approach application.

5. Conclusions

This article solves the problem of determining the spatial position of the operator using keypoints with a multi-camera sensor system. The possibility of the proposed approach application to solve the problems of collaborative robotics has been demonstrated. The advantage of the algorithm is the use of a multi-chamber system that allows you to attach points to the local coordinate system of an industrial robotic complex. The obtained data will further be used in planning the trajectory of the robot to safely perform joint operations in the same workspace. As additional modifications, the possibility of installing a lidar to clarify the coordinates of the edges of the skeleton is being considered.

A research perspective is the development of an algorithm to predict the actions of the operator in the workspace and detect hazardous situations and possible intersections in the trajectories of a collaborative robot.

Author Contributions: Conceptualization, Y.I., S.Z. and M.G.; methodology, Y.I. and S.Z.; software, S.Z. and D.G.; validation, S.Z., M.G. and S.S.; writing—original draft preparation, Y.I. and S.Z.; writing, review and editing, Y.I. and D.G. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the Russian Science Foundation (project № 22-71-10093).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ivanov, Y. S.; Kosichkov, A.O. Recognition and Control of the Athlete's Movements Using a Wearable Electronics System. In Proceedings of the International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), Vladivostok, Russia, 6–9 October 2020; pp. 1–5.
2. Amosov, O.S.; Amosova, S.G.; Ivanov, Y.S.; Zhiganov, S.V. Using the deep neural networks for normal and abnormal situation recognition in the automatic access monitoring and control system of vehicles. *Neural Comput. Appl.* **2020**, *33*, 3069–3083. [CrossRef]
3. Amosov, O.S.; Amosova, S.G.; Ivanov, Y.S.; Zhiganov, S.V. Using the Ensemble of Deep Neural Networks for Normal and Abnormal Situations Detection and Recognition in the Continuous Video Stream of the Security System. *Procedia Comput. Sci.* **2019**, *150*, 532–539. [CrossRef]

4. Sigal, L. *Human Pose Estimation*; Ikeuchi, K., Ed.; Computer Vision; Springer: Boston, MA, USA, 2014.
5. Gril, L.; Wedenig, P.; Torkar, C.; Kleb, U. A Tensor Based Regression Approach for Human Motion Prediction. 2022. Available online: https://arxiv.org/abs/2202.03179 (accessed on 15 June 2022).
6. Huck, T.P.; Ledermann, C.; Kröger, T. Testing Robot System Safety by Creating Hazardous Human Worker Behavior in Simulation. *IEEE Robot. Autom. Lett.* **2022**, *7*, 770–777. [CrossRef]
7. Makris, S.; Karagiannis, P.; Koukas, S.; Matthaiakis, A. Augmented reality system for operator support in human–robot collaborative assembly. *Cirp-Ann.-Manuf. Technol.* **2016**, *65*, 61–64. [CrossRef]
8. Park, J.S.; Park, C.; Manocha, D. I-planner: Intention-aware motion planning using learning-based human motion prediction. *Int. J. Robot. Res.* **2019**, *38*, 23–39. [CrossRef]
9. Zhang, J.; Liu, H.; Chang, Q.; Wang, L.; Gao, R.X.Recurrent neural network for motion trajectory prediction in human-robot collaborative assembly. *CIRP Ann.* **2020**, *69*, 9–12. [CrossRef]
10. Cheng, Y.; Sun, L.; Liu, C.; Tomizuka, M. Towards Efficient Human-Robot Collaboration With Robust Plan Recognition and Trajectory Prediction. *IEEE Robot. Autom. Lett.* **2020**, *5*, 2602–2609. [CrossRef]
11. Xu, Y.; Zhang, J.; Zhang, Q.; Tao, D. ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation. 2022. Available online: https://arxiv.org/abs/2204.12484 (accessed on 6 July 2022).
12. Papers with Code. MPII Human Pose Benchmark (Pose Estimation). 2022. Available online: https://paperswithcode.com/sota/pose-estimation-on-mpii-human-pose (accessed on 15 July 2022).
13. Carraro, M.; Munaro, M.; Burke, J.; Menegatti, E. Real-time marker-less multi-person 3D pose estimation in RGB-Depth camera networks. In *Intelligent Autonomous Systems 15*; Springer: Berlin/Heidelberg, Germany, 2018.
14. GitHub, Inc. Open PTract v2. 2022. Available online: https://github.com/OpenPTrack/open_ptrack_v2 (accessed on 28 July 2022).
15. Elhayek, A.; de Aguiar, E.; Jain, A.; Thompson, J.I.; Pishchulin, L.; Andriluka, M.; Bregler, C.; Schiele, B.; Theobalt, C. MARCOnI—ConvNet-Based MARker-Less Motion Capture in Outdoor and Indoor Scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 501–514. [CrossRef] [PubMed]
16. Lora, M.; Ghidoni, S.; Munaro, M.; Menegatti, E. A geometric approach to multiple viewpoint human body pose estimation. In Proceedings of the 2015 European Conference on Mobile Robots (ECMR), Lincoln, UK, 2–4 September 2015; pp. 1–6.
17. Takahashi, K.; Mikami, D.; Isogawa, M.; Kimata, H. Human Pose as Calibration Pattern: 3D Human Pose Estimation with Multiple Unsynchronized and Uncalibrated Cameras. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 1856–18567.
18. Papers With Code. Pose Estimation Subtasks. 2022. Available online: https://paperswithcode.com/area/computer-vision/pose-estimation (accessed on 3 August 2022).
19. COCO Dataset. COCO–Common Objects in Context. 2022. Available online: https://cocodataset.org/#home (accessed on 5 August 2022).
20. Ferrari, V.; Marín-Jiménez, M.J.; Zisserman, A. Progressive search space reduction for human pose estimation. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, 23–28 June 2008; pp. 1–8.
21. Andriluka, M.; Pishchulin, L.; Gehler, P.V.; Schiele, B. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3686–3693.
22. Brown, D.C. Close-Range Camera Calibration. *Photogramm. Eng.* **1971**, *37*, 855–866.
23. Zauss, D.; Kreiss, S.; Alahi, A. Keypoint Communities. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 11037–11046.
24. Kreiss, S.; Bertoni, L.; Alahi, A. OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 13498–13511. [CrossRef]
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
26. Ma, N.; Zhang, X.; Zheng, H.; Sun, J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.