Escuela Superior Politécnica del Litoral

# Human Pose Estimation based in Deep Learning Techniques from Multi-view Environments

A dissertation submitted by **Jorge Charco Aguirre** at Escuela Superior Politécnica del Litoral to fulfil the degree of **Doctor in Applied Computer Science**.

Guayaquil - Ecuador, March 1, 2023

| | |
|---|---|
| Director | **Dr. Boris X. Vintimilla**<br>Escuela Superior Politécnica del Litoral, Ecuador |
| Co-Director | **Dr. Angel D. Sappa**<br>Escuela Superior Politécnica del Litoral, Ecuador<br>Centre de Visió per Computador, España |
| Thesis committee | **Dr. Sergio Velastin**<br>Universidad Carlos III de Madrid, España<br>Queen Mary University of London, England |
| | **Dr. Wilbert Aguilar**<br>Universidad de las Fuerzas Armadas, Ecuador |
| | **Dr. Enrique Pelaez**<br>Escuela Superior Politécnica del Litoral, Ecuador |
| | **Dr. Miguel Realpe**<br>Escuela Superior Politécnica del Litoral, Ecuador |

This document was typeset by the author using LaTeX 2$_\varepsilon$.

The research described in this book was carried out at the Escuela Superior Politécnica del Litoral.

But blessed is the one who trusts in the Lord,
whose confidence is in him.
They will be like a tree planted by the water
that sends out its roots by the stream.
It does not fear when heat comes;
its leaves are always green.
It has no worries in a year of drought
and never fails to bear fruit.

Jeremiah 17:7-8 NIV

To Almighty God and my family.

# Acknowledgements

First of all I want to thank God for giving me health during these years, wisdom, constancy and strength to complete the proposed goals. To my wife, my children and parents, who supported me all the time and could understand that I could not always be or spent more time with them.

A special thank to my director and co-director, Dr. Boris Vintimilla and Dr. Angel Sappa, who have been the key to me becoming a competent researcher, having patience to teach me everything necessary to fulfill the objectives of my thesis. Thanks for introduced me in the scientific field through of a novel problem and their advice to how to tackle these challenging problems. Thanks for being my mentors and now my friends, leading and supporting me along the way to successfully completed PhD program.

I would like to thank all people who conform the Center for Research, Development and Innovation in Computing Systems (CIDIS), to made my stay pleasant and comfortable during this time. I would also like to extend thanks to my great thesis committee: Dr Sergio Velastin, Dr. Wilbert Aguilar, Dr. Enrique Pelaez and Dr. Miguel Realpe.

During these years, I have gained many friends, always having interesting talks during the coffee or after of any scientific events with Rafael Rivadeneira, Sebastian Quevedo, Eduardo Cruz, Roberto Jácome and Marjorie Chalen. I would also like to thank to Henry Velesaca for your help during this time and whom I consider a friend.

I am especially grateful to SENESCYT that supported me with the scholarship for the PhD program.

# Abstract

The field of computer vision has had great progress during the last decades due to the advancement of hardware computing power, specifically in Graphics Processing Unit (GPU). Although GPSs have been design for the gaming industry, also have been useful to design powerful algorithms for solving some problems such in this field research such as segmentation, detection, structure from motion, camera calibration and many others. These problems are common in applications like autonomous driving, robot navigation, video surveillance for human tracking, action recognition or human body pose estimation. Some techniques has been applied to tackle these tasks, one of the most widely used deep learning based technique, is the convolutional neural network (CNN) due to the power for features extraction in images.

This dissertation presents a series of CNN-based techniques applied to images to tackle the camera pose and human body pose estimation problems from multi-view environments. For the camera pose estimation, two approaches based on Siamese architecture have been proposed to estimate the camera pose—extrinsic parameters. The first approach takes as input a set of pairs of real-images, which should have a minimum overlapping to ensure that the pairs of images have common features. However, due to few real-image datasets available to the camera pose estimation from multi-view scenarios, a second approach is proposed. It consists of domain adaptation strategy, including the generation of different virtual scenarios by using a special 3D simulation software. The strategy is used to take advantage of transferring of learned knowledge from these virtual scenarios to real-world scenarios. For the human body pose estimation problem, two approaches are also proposed. The first, an architecture based on convolutional neural network, which leverages the estimated extrinsic parameters to establish the relationship between different cameras into the multi-view scheme. It has allowed to estimate the human body pose using information from different points of view, and thus, to solve the challenging problem of the self-occlusion in human pose estimation due to the natural body pose. A second approach for the human body pose estimation problem has been also proposed. It uses attention modules to detect body joints. However, unlike to the first approach, this new approach does not take into account the extrinsic parameters between different cameras of the multi-view scheme, instead, the position and orientation of bones of human body are used as additional information to tackle the problem of self-occlusion of human body joints. The accuracy of these estimations is important to avoid possible false alarms in behavioral analysis systems of smart cities as well as applications for physical therapy, safe moving assistance for the elderly among other.

Keywords: Convolutional neural network, camera calibration, relative camera pose, siamese architecture, multi-view environments, deep learning, domain adaptation, human body pose, attention modules.

# Resumen

El campo de la visión por computador ha tenido un gran progreso durante las últimas décadas debido al avance de la potencia de procesamiento del hardware, específicamente en la Unidad de Procesamiento Gráfico (GPU). Aunque las GPUs han sido diseñadas para la industria del juego, también han servido para diseñar potentes algoritmos para resolver algunos problemas en este campo de investigación tales como la segmentación, la detección, la estructura a partir del movimiento, la calibración de la cámara y muchos otros. Estos problemas son comunes en aplicaciones como la conducción autónoma, la navegación de robots, la videovigilancia para el seguimiento de personas, el reconocimiento de acciones o la estimación de la pose del cuerpo humano. Se han aplicado algunas técnicas para abordar estas tareas, una de las técnicas más utilizadas basadas en el aprendizaje profundo, es la red neuronal convolucional (CNN) debido a la potencia para la extracción de características en las imágenes.

Esta tesis presenta una serie de técnicas basadas en CNN aplicadas a imágenes para abordar los problemas de estimación de la pose de la cámara y del cuerpo humano a partir de entornos multivistas. Para la estimación de la pose de la cámara, se han propuesto dos enfoques basados en la arquitectura siamesa para estimar los parámetros extrínsecos de la pose de la cámara. El primer enfoque toma como entrada un conjunto de pares de imágenes reales, que deben tener un solapamiento mínimo para asegurar que los pares de imágenes tienen características comunes. Sin embargo, debido a los pocos conjuntos de datos de imágenes reales disponibles para la estimación de la pose de la cámara en escenarios multivista, se propone un segundo enfoque. Este consiste en una estrategia de adaptación del dominio, que incluye la generación de diferentes escenarios virtuales mediante un software especial de simulación 3D. La estrategia se utiliza para aprovechar la transferencia del conocimiento aprendido de estos escenarios virtuales a los escenarios del mundo real. Para el problema de estimación de la postura del cuerpo humano, también se proponen dos enfoques. El primero, una arquitectura basada en una red neuronal convolucional, que aprovecha los parámetros extrínsecos estimados para establecer la relación entre las diferentes cámaras en el esquema multivista. Esto ha permitido estimar la pose del cuerpo humano utilizando información de diferentes puntos de vista, y así, resolver el desafiante problema de la auto-oclusión en la estimación de la pose humana debido a la pose natural del cuerpo. También se ha propuesto un segundo enfoque para el problema de la estimación de la pose del cuerpo humano. Este utiliza módulos de atención para detectar las articulaciones del cuerpo. Sin embargo, a diferencia del primer enfoque, este nuevo enfoque no tiene en cuenta los parámetros extrínsecos entre las diferentes cámaras del esquema multivista, sino que se utiliza la posición y la orientación de los huesos del cuerpo humano como información adicional para abordar el problema de la auto-oclusión de las articulaciones del cuerpo humano. La precisión de estas estimaciones es importante para evitar posibles falsas alarmas en los sistemas de análisis del comportamiento de las ciudades inteligentes, así como en aplicaciones de fisioterapia y asistencia para el desplazamiento seguro de personas mayores, entre otras.

Palabras claves: Redes neuronales convolucionales, calibración de cámaras, pose relativa de la cámara, arquitecturas siamesas, ambientes de múltiples vístas, aprendizaje profundo, adaptación de dominio, pose del cuerpo humano, módulos de atención.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

Computer Vision is a research field where the different algorithms are used to do the tasks that the human visual system can do. For this, the ability to understand the images and videos context should be learn by machines. The analysis and interpretation of visual information of an image, which is a projection from real world onto a 2D plane has been a challenging problem for years. The developed solutions for image understanding have allowed machine learns the context of the scene through of numerical or symbolic information. However, the representation of the real world captured in the image could be affected by some factors of camera such as lens, distortion or focal length, including external factors as poor illumination in the scene. In order to tackle this problem, the camera calibration has become an essential step in some computer vision applications since it helps to find the internal and external camera parameters, known as intrinsic (distortion and focal length) and extrinsic (camera position and orientation) parameters. Some computer vision applications such as mobile robots, driving assistance, human pose estimation, augmented reality, just to mention a few, use the automatic camera calibration as their principal process since they allow to obtain the relationship between the cameras of the scene and the world coordinate system [31, 56, 74, 85].

The computer vision applications have had great advances due to the hardware computing power, Graphics Processing Unit (GPU) and large amount of data available to train algorithms. The usage of GPU has allowed to design powerful computer vision algorithms, in particular deep learning based architectures, which use massive amounts of data for the learning process. The existence of massive amounts of data has become a critical factor to train these architectures. Considering this, some virtual simulator tools have been used like alternative to generate these data and thus, can use these architectures.

A type of deep learning architecture used for visual analysis of images is Convolution Neural Network (CNN). During the last years, these architectures have shown outstanding results, even improving the state-of-the-art in different computer vision tasks such as classification, object detection, segmentation, super-resolution, just to mention a few tasks [51, 86, 119]. In order to leverage the power of CNN, different algorithms have been proposed to tackle the camera calibration and Human Pose Estimation (HPE) problems. Basically, the human

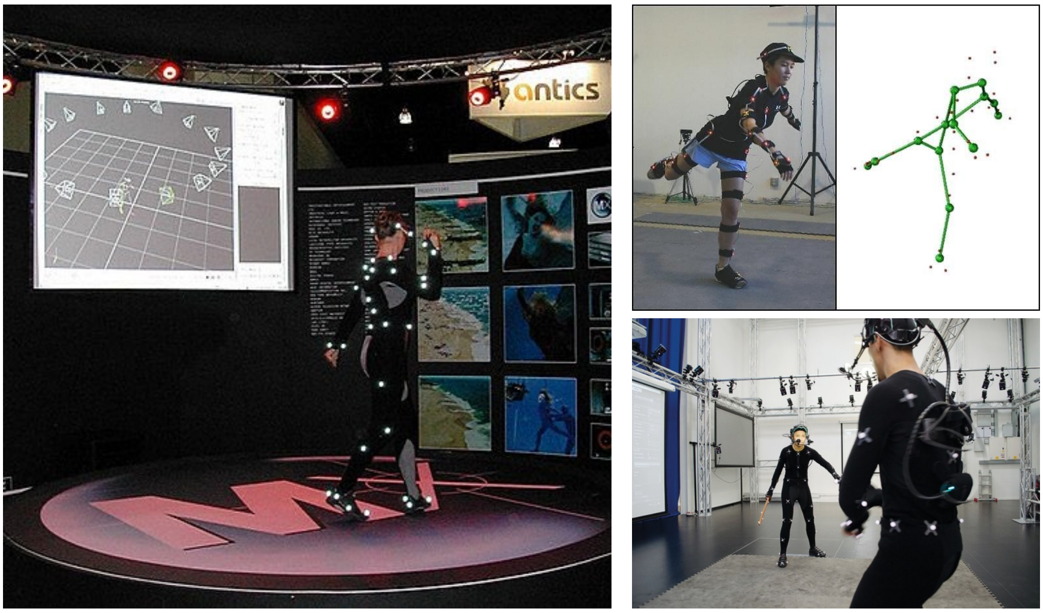


Figure 1.1: Optical markers to estimate the human pose in controller environments [79].

pose term refers to the position that the human body could take while an action or activity is performed; and similar way like people recognize the position of human body (i.e., looking at the orientation and location of each part of the human body), the algorithms estimate it through to the human body joints.

Nowadays, optical markers attached to body's part are used to estimate the human pose, getting high accurate (see Fig. 1.1). However, this solution can be used only in controlled environments (i.e., experimental labs or production studios), and are very expensive [21]. Hence, these optical markers are not suitable for applications in uncontrolled environments, for instance, sports analysis, video surveillance or medical assistance, just to mention a few applications. In order to tackle this problem in outdoor environments, some methods have been proposed without the need that people wear a special suit with optical markers to estimate the human pose. These methods, which are Markerless approaches, use RGB images as main source of information, which can be captured from any commercial camera, and are the most used in computer vision tasks such as classification, human tracking, action recognition, among other. Other source of information used by Markerless approaches are depth images. However, these require specific hardware able to capture depth information (e.g., Microsoft Kinect [129], Leap Motion [65] or stereo vision systems [78]). Although these hardware are not expensive, they have a distance limitation of about 4 meters to acquire images.

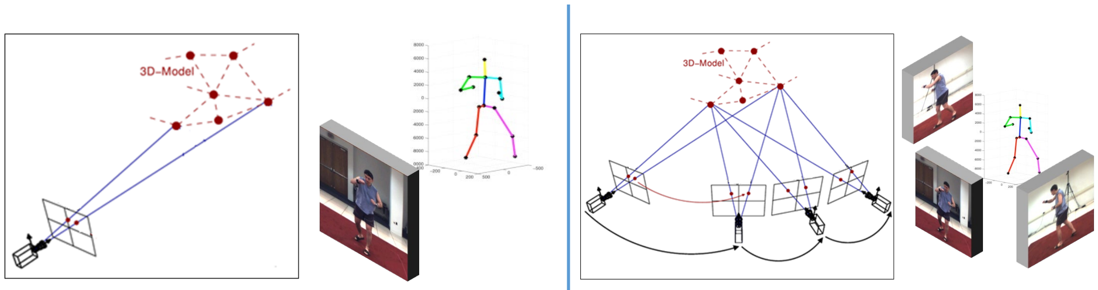
Considering the limitations mentioned above, Markerless approaches based on RGB images, are mostly used to detect human body joints, which are later on linked to define the body pose. The markerless-based approaches should tackle several challenging situations that could affect the performance of the algorithms to find the human body joints such as change or poor illumination in the image, large variation in clothing as well as different human body posture. Although there have been advances and robust solutions have been proposed for the HPE, mainly when all body joints can be detected; it becomes a challenging problem when these joints are occluded due to the natural human body pose in the scene, which is something common in monocular vision system, and that have not been completely solved. An alternative to tackle this problem is a multi-view approach, which has been already explored to the region occlusion problem in tasks such as 3D-reconstruction, autonomous driving, object detection [39, 91, 108, 131]. Multi-view approaches can simultaneously capture the human body parts from different cameras and points of views. Thus, any occluded joint in one camera can be observed by some of the other cameras from other point of view (see Fig. 1.2).

## 1.1 Problem Statement

As mentioned above, this thesis is focused on the human pose estimation in multi-view environments, before tackling the human pose estimation problem, the different cameras of the acquisition system need to be referred to the same coordinate system through a camera calibration process, ideally in an automatic way.

The automatic camera calibration is an important step in several computer vision tasks and can be tackled both in the single-view as well as in the multi-view scheme. The multi-view camera calibration process is considered in this framework of this thesis. For this problem, we assume that two o more RGB cameras capture the scene from different points of view, and an overlapping restriction is imposed in order to ensure that common features could be found in the images captured by the cameras. Additionally, the objects in the scene could be, either static or moving objects, and they should have rich textures.

The HPE is the second problem tackled in this thesis. It is an important research field, which has been studied for years. It can help in areas such as action recognition, medical assistance, sport motion analysis, among other. Although the monocular approach has been widely-used to solve the HPE, several factors, such as occluded joints, complex poses or objects that may partially occlude the human body, affect the estimation. Hence, the multi-view framework is considered. Like in the camera calibration some constraints are also imposed (i.e., two RGB cameras and overlapping between them); additionally, the constraint that the scene contains just a single object is considered. The scene should have rich textures and objects with different shapes in the background, which is required for that the automatic camera calibration process takes advantage of the relevant features in the images. Although the scene could contain more than one persons, the proposed approach will take only one person at a time to estimate its pose.



(a) Single-View

(b) Multi-View

Figure 1.2: General scheme of single-view and multi-view approaches to estimate the human pose.

## 1.2 Challenges

### Single-View

Aside of common challenges in RGB images captured like blurring, illumination changes, poor lighting, among others, a specific challenge in HPE is the occlusion of joints, mainly when the scene is captured by a single camera (see Fig. 1.2) . The human body can move freely in real-life scenarios and easily the natural pose of body can generate self-occlusions of certain body's part. The cloth styles and colors add also difficult to the problem. In a RGB image, the depth information of each body joint is lost, making more complex to recover the human pose. Additionally, constraint of skeleton models to reduce the search space are not considered in the current work, doing more complex to find the body joints using just RGB images.

### Multi-View

Although, in general, the challenges are similar to single-view respect to RGB images, it has the possibility of disambiguating the occluded body's parts due to the fact that the scene is simultaneously captured from different points of view (see Fig. 1.2). However, since the scene is captured from different cameras and points of views, finding the relationship between all cameras is necessary in order to put in the same reference system all feature points; hence, take advantage of the redundant information to associate the occluded regions. The pair of images should meet the overlap restriction of at least 60% to ensure that the RGB images share information.

## 1.3 Objectives

This thesis focuses on the exploration and design of deep learning based techniques to the human pose estimation problem. The objective is to develop a robust solution to be used in multi-view environments without any constraints. In such non-constrained multi-view environments, the camera pose needs to be automatically estimated using just the provided images.

In order to tackle the HPE problem in multi-view environments, the relative relationship between cameras of the scene is used, since that it could help us to improve the accuracy in the body joints, mainly those occluded joints, which could be due to the natural body pose; among the most important research objectives we have:

- Evaluate the importance of synthetic image datasets within the training process of the proposed architectures, to solve Camera Pose Estimation (CPE) problem in multi-views environments.
- Design new deep learning architectures considering multi-view scenarios to solve human body parts occlusion problems when human pose is estimated.
- Determine if the use of relative camera pose within training process of the proposed architectures in multi-views environments benefit the human pose estimation when body parts are occluded.

## 1.4 Research Questions

The following three research questions have been formulated with respect to the objectives of this thesis:

- How to tackle the camera pose estimation problem in multi-view environments when there is a lack of large datasets for algorithm training?
- How the relative camera pose from multi-view environments could help to estimate occluded human body parts?
- Is it possible to increase the accuracy of human pose estimation if multi-view environments are considered?

## 1.5 Structure of the thesis

This thesis is organized in five chapters. The first one presents an introduction, and it is focused on explaining the context of the research and how the CNN can help to improve the performance of the algorithms to tackle the camera calibration and human pose estimation problems. Furthermore, the problem statement, the objectives and research questions of the thesis are presented.

Chapter 2 presents the state-of-the-art for understating the camera pose and human pose estimation problems. For the camera pose, the classic methods and new deep learning architectures proposed in the literature are studied to know the different solutions found about this problem, and how the multi-view approaches have helped to find the position of all cameras. The used single-view and multi-view approaches for HPE problems are analyzed, as well as how the camera calibration into of this scheme have been used to take advantages of redundant information. The state-of-the-art of attention modules are also revised, including an overview of the different models applied to the HPE. Finally, a benchmark of different datasets available for camera pose and human pose estimation are showed.

Chapter 3 covers the CPE problem. The design of a Siamese network architecture is proposed to leverage the multi-view scheme. Also virtual environments are used to solve the lack of large datasets of real images required for training the proposed models, considering that they should be focused on multi-view schemes. Furthermore, a domain adaptation strategy is proposed to avoid the need of having these large real-images datasets for training process. The experimental results are presented, including different comparisons to show the performance of proposed model.

Chapter 4 covers the HPE problem. New models by using attention modules are proposed. Specific information about joints position is used to guide these proposed models during the training process and thus, improve the performance. They are based in a single-view, which can be used as backbone in the proposed multi-view scheme. The CPE mentioned in Chapter 3 is required to put all images in the same reference system. Different comparisons are showed, including cases of complex poses to evaluate the performance of proposed models.

Finally, Chapter 5 contains the general conclusions of problems tackled in the thesis. Additionally, future research lines related to this thesis are described.

# 2 Related Work

The thesis studies the HPE problem in a multi-view environment and how the camera calibration is an important previous step to estimate the relative camera pose (i.e., position and orientation) between all cameras, and thus, take advantages of the redundant information captured by each one of them. This chapter presents the state-of-the-art on CPE and HPE problems from classic methods to new techniques using CNN.

## 2.1 Camera Pose Estimation

A problem that is tackled during the thesis is the camera calibration since that is used in the HPE from multi-view environments. The camera calibration allows to find the intrinsic (i.e., fundamental camera matrix, focal length and distortion) and extrinsic parameters (i.e., relative rotation and translation) of a camera (see Fig. 2.1).

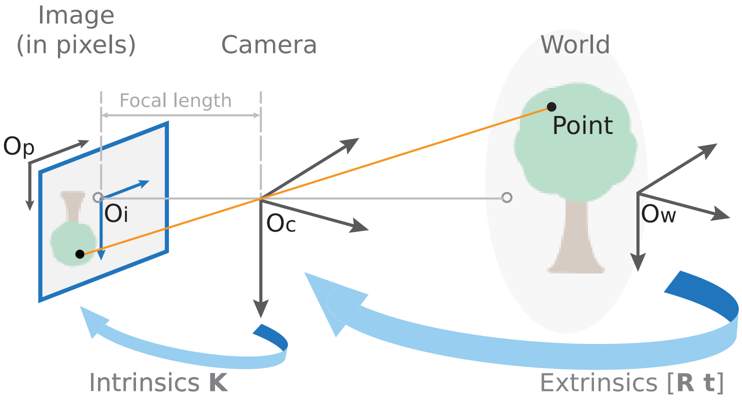
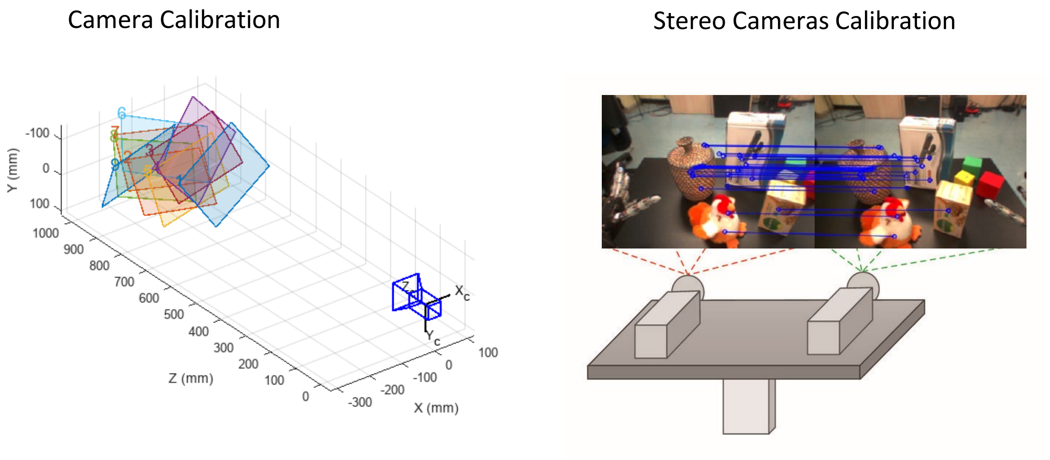


Figure 2.1: Camera calibration: intrinsic and extrinsic parameters [66].



Camera Calibration

Stereo Cameras Calibration

Figure 2.2: According to [66]. *(Left):* Camera calibration using chessboard. *(Right):* Stereo camera calibration using SIFT algorithm.

In the current work only the extrinsic camera parameters estimation are analyzed referred to as camera pose estimation. For CPE process, it is necessary to get a relationship between 3D world points and 2D image points on the camera plane. Several factors affect the camera calibration process such as poor illumination, low resolution and few image features.

This section starts presenting classical approaches proposed in the literature and then recent deep learning techniques are also reviewed.

### 2.1.1 Classic Methods

During the last years different approaches have been proposed in the literature for camera calibration. In [28], the authors have proposed a method that does not require a calibration object with a known 3D shape. Instead, only points of interest matches from image sequences are necessary. These matched points could be tracked in the images as the camera moves. For this, the camera calibration is performed in two steps; in this first one, the epipolar transformation is found through a generalization of the essential matrix. In the second one, Kruppa equations, proposed by [67], are used to link the epipolar transformation to the image to estimate the camera parameters. In order to solve these coefficients of the images, at least three movements are required. Another method where at least three images of a scene are required to estimate the camera parameters, is proposed by the author in [35]. These images are captured from the same point in space but with different camera's orientation, making more easier finding the same relevant features in all image, in contrast to the images taken with a moving camera, avoiding the problem of occlusion and illumination in the images. The method is based on the image's content and it does not require a priori assumptions of calibration values; additionally, how all images are taken from the same point in space but with different rotation, epipolar structure for camera calibration process is not required.

Unlike methods that do not require a calibration object, the feature detection and matching based algorithms such as SURF, ORB, SIFT [3, 72, 87], have become an important point for classic approaches. Some works have used these algorithms for camera calibration process. In [62], the authors propose to use a pattern calibration (e.g., plane chessboard based) to calibrate each single view, and then use the SIFT algorithm to calibrate the stereo cameras after getting two camera parameters (see Fig. 2.2). The stereo cameras allow to simulate human binocular vision and therefore gives it the ability to perceive depth. In order to determine the fundamental matrix, which describes the geometric relationship between corresponding points in stereo cameras, is necessary to know the matching features points between both images; for this, the SIFT method is used. After the intrinsic parameters and fundamental matrix are combined for getting the essential matrix (i.e., a 3×3 matrix that relates corresponding points between images assuming that the cameras are calibrated); and thus, using the singular value decomposition it can obtain the translation and rotation matrix. Other algorithms such as Structure from Motion (SfM) have taken advantage of SIFT method since it allows to identify invariant features to scale, rotation or change of illuminations in the images. The authors in [1, 94] propose to use SfM algorithms for 3D-reconstruction from image sets or video sequence. The first step of the algorithm is correspondence search, which finds scene overlap in the images. Then, the local features extraction is performed using SIFT method, and together with the camera parameters, the camera pose is estimated. Finally, 3D points are triangulated and optimized in a bundle adjustment scheme.

These methods have a critical factor that is the number of feature points found and matched between the images, being an important factor to get better accuracy of their results. Hence, they have a low performance when the scene contains poor illumination or lack of texture.

### 2.1.2 Deep Learning Architectures

The neural networks have been the best major advance for learning process of the machine in most computer vision tasks since they allow to learn representative features from input data. Considering the powerness of neural networks, the authors in [71] have proposed a procedure for camera calibration using neural networks of the type of back propagation perceptron and thus, to learn the relationship between world coordinate system and the 2D-coordinates of the image plane. An advantage of the proposed method is that it does not require to know the camera parameters, being robust enough for different cameras, even different focal lengths. This approach takes as input the points of images in both cameras, and the outputs are the corresponding 3D space coordinates of each point. A similar approach has been also proposed in [22], where a neural network is used for the stereo calibration process. In order to avoid complicated mathematical modeling, a projection and back-projection are considered for direct mapping from 3D world coordinates to 2D-image coordinates respectively, where a multilayer neural network is used since they are able of approximating a continuous nonlinear function. However, a drawback of this approach is to find the optimal number of nodes in

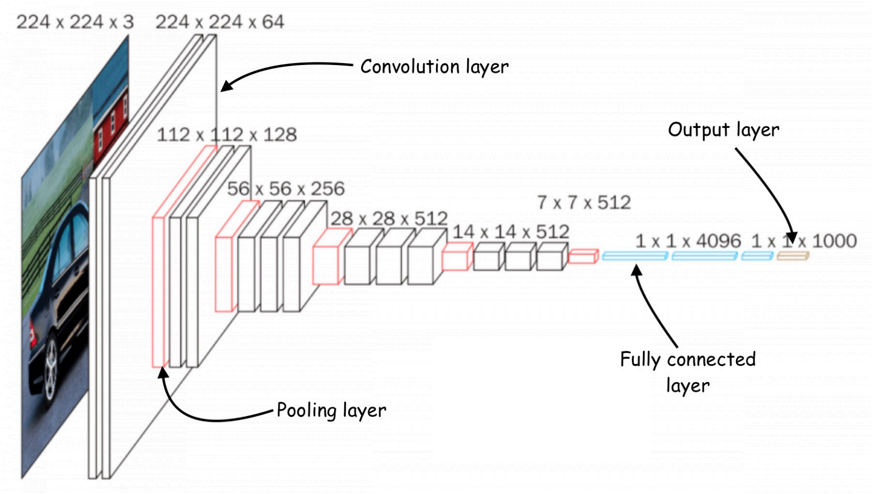


Figure 2.3: CNN architecture to tackle computer vision tasks [99].

hidden layers to avoid the overfitting or underfitting.

On the other hand, CNNs are also a type of neural network, which allows to extract features directly on images, avoiding the manual feature extraction (see Fig. 2.3). During last years, these neural networks have been used to solve some computer vision tasks such as classification, segmentation, tracking object, super-resolution among other, showing outstanding results with respect to classic approaches [51, 86, 119]. Considering this, the camera calibration problem could be tackled using the power of CNNs, with the advantage that will not require the usage of a specific calibration pattern. Dynamic objects (i.e., pedestrians, cars) that could be into the scene will become a problem, due to the fact they may change their position while the images are captured, making that certain region of scene with features points could be occluded at consecutive frames.

Focusing on this problem, the authors in [109] have proposed an architecture that combines the powerness of neural networks with the geometric principles of images that are captured from a moving camera. The approach has two modules: motion estimation and depth estimation. For the first module, an architecture called Flow-SE3 is used to estimate camera motion, which takes depth estimated from image sequence as input, to estimate dense 2D correspondence between pairs of frames. While for the depth estimation module, it takes camera motion as input, and formulates multi-view stereo reconstruction as a single feed-forward network to predict a dense depth map. For the optimization process, the global and keyframe pose optimization are used. The first, uses all pairs of frames to define the objective function, while the second one, selects a frame to be the keyframe and only computes the error terms between the keyframe and each of the other frames. The approach alternates the depth and motion modules creating a mutual dependence during the training process. Hence, when the estimated depth is more accurate, then the camera pose is also more accurate. Likewise, as camera pose is more accurate, then the depth module can estimate more accurate depth.

The authors in [54] have proposed a CNN, which uses RGB images to regress the 6-DOF camera pose. It is robust to change of illumination, motion blur and different camera intrinsic parameters. The problem of this model is the necessity of large amount of images for the training process. In order to overcome it, the learned knowledge from other tasks is used, which is known as transfer learning. It is a strategy that helps to reach a fast convergence with a small error. The proposed approach uses a 3-dimension vector to represent the position of camera and a quaternion to represent the orientation. Euclidean Distance is used as a loss function, including a scale factor for keeping the value expected for position and orientation errors at a similar scale. This scale factor is fine-tuned using a grid search. An update of previous approach has proposed in [52], where a novel loss function based on geometry and reprojection error for camera pose estimation is used. For avoiding the scale factor in loss function, the homoscedastic uncertainty [53] is used as a measure of uncertainty to learn a weighting between the position and orientation objective functions, which does not depend on the input data. For case based on reprojection error, 3D points in the scene projected are obtained if the scene's geometry is known, making that the weight between position and orientation can be adjusted during the training process.

A method based on people observation in the input video, with known focal length of the static camera, has been proposed in [95]. The approach does not require any special calibration template to estimate camera parameters. A synthetic dataset is taken on a ground plane with people standing on it and camera placed above with different camera parameters, which is used for training process and after generalized to real-world environments. The bounding boxes of detected human head and focal length are required as input to predict camera extrinsic parameters.

Other approach to estimate the intrinsic camera parameters (i.e., focal length and distortion parameters) from a single image of general scenes, has been proposed in [4], which uses a novel fully automatic deep learning-based approach. A large image collection of panoramas available on the internet is leveraged to generate a large-scale dataset. The images are linearly mapped onto an unit sphere and then, using a virtual camera with the desired values for focal length and distortion parameters, a new synthetic image is created. One of the variants of this proposal uses a sequence of two joint networks, where the output of first part is the focal length parameter, and together with the input image, feed to the second part of architecture to estimate the distortion parameter. The authors in [46] have proposed to use the geometric and photometric information to tackle the problem of LiDAR-Camera calibration. For this, a novel architecture based on 3D spatial transformers is used. The proposal requires as input a LiDAR point cloud, the monocular RGB image and the matrix of camera calibration K (i.e., intrinsic parameters) to predict the extrinsic calibration parameters that maximize the geometric and photometric consistency of the input RGB images and point clouds.

In [61], the authors have proposed to use a Recurrent Convolutional Neural Network

(RCNN), including Long-Short Term Memory (LSTM) for accurate pose prediction. A CNN is used to extract the image sequence features, RCNN estimates the relative and global pose from extracted image features, and finally fully-connected fusion layers fuse the global and relative pose to obtain the translation and rotation (quaternions) between pairs of images sequence. The architecture parameters are optimized using the cross-transformation consistency, which is employed to enforce the temporal geometry consistency of the consecutive frames, and Mean Square Error (MSE) between the predicted pose and ground truth.

The previous approaches have the challenging problem of occluded regions of interest in scene, which could be present in single-view approaches, and that have not been completely solved. The multi-view approach could be used to tackle this problem, since the scene can be simultaneously captured from different points of view by different cameras, considering a minimum overlap of regions between the captured images.

In this direction, just few works have been proposed to solve the camera pose estimation problem in multi-view environments. For instance, in [70], the authors have proposed to use a Siamese CNN architecture that takes as input RGB images from both cameras to estimate the relative camera pose. The AlexNet architecture is used as base network for both network branches with shared weights. The weights of proposed architecture are initialized with the learned weights from classification task on *ImageNet* and *Places* datasets [88, 139]. For training process, five landmarks are considered (Montreal Notre Dame, Piccadilly, Roman Forum, Vienna Cathedral and Gendarmenmarkt), while *DTU Robot Image* dataset for validation. Respect to the first dataset, the ground truth is computed by applying the SIFT keypoint detector and SfM technique. In the second dataset, *DTU Robot images*, the ground truth is obtained by attaching the camera to a precisely positioned robot arm. A scale factor, like in [54], is used to balance the translation and rotation errors. Euclidean distance is used as a loss function. Similar to the mentioned above, the authors in [26] have also proposed a Siamese Network, which uses the GoogLeNet as based for the network where three variants are included (i.e., a parameter-free module, a parameter-free module with additional losses and a relative pose regressor based on FC-layers), to predict relative camera pose between two cameras. The loss function is also used like in [54], where a scale factor helps to balance the loss values between the translation and rotation. The output of the network proposed by [54], is a 7-Dimensional vector where the translation is represented by a 3-Dimensional vector (x, y, z) and the rotation is represented as a quaternion. For training and validation processes, the *Cambridge Landmark* [54] dataset is used.

In [14], the authors present a novel architecture called DirectionNet for relative pose estimation. The architecture contains two parts: in the first one, a Siamese branch is used to encode image pairs, and obtain as output image embedding, while in the second one, a spherical encoder maps the obtained image embedding to a spherical distribution. The network estimates discrete distributions over the 5-Dimensional relative pose space using a novel parameterization to make the estimation problem tractable. Furthermore, the spherical distribution is used to compute sets of 3D direction vectors, and thus, estimate the 3D rotation and translation directions. However, when the relative rotation between the cameras is large, an intermediate frame with a larger Field of View (FoV) and increased resolution are necessary to be used. Distribution loss that provides dense supervision and spread loss that penalizes the spherical “variance” are used like part of loss function. Additionally, the direction loss, which considers similarity between two 3D vectors, is also included.

Likewise, an architecture that estimates 6-DoF pose from an image and a 3D model is proposed in [90]. The model is based on the direct alignment of multi-scale deep features. The features are extracted from query and reference images, including pixel-wise confidences. In order to align corresponding features according to the 3D points, the Levenberg-Marquardt optimization is used, which also consider the confidence as guide in the learning process. The goal of the optimization is to minimize the difference in appearance between both images (query and reference images), and thus, to obtain major accuracy in the pose (R, t). Two datasets are used for the training process. The first one is the *MegaDepth* dataset [58], which is composed of popular image landmarks around the world, while the second one, *Extended CMU Seasons* dataset [92], contains a collection of image sequences of urban and rural environments. The proposed model is evaluated using Cambridge Landmarks and 7-Scenes datasets [54, 96]. The authors in [34], have proposed different methods using Affine Correspondences (ACs) for solving the relative pose problem of a multi-camera system. ACs have more information than a point correspondence; hence, few ACs are required to tackle the problem. These methods can be used within RANSAC for outlier removal and initial motion estimation. Unlike of previous methods, they do not use CNN to estimate the relative pose, instead they use three different methods: the first one, Grobner basis method, the second one, based on planar motion and finally, other method with known vertical direction. In the latter one, an Inertial Measurement Unit (IMU) is attached to the multi-camera system with known relative rotation between IMU and the reference frame.

## 2.2 Human Pose Estimation

Regarding the HPE problem, it is defined as the problem of body joints localization (i.e., elbow, wrist, shoulder among other) to get the human pose from a given image or video. As traditional method, the optical markers attached to body parts are used to accurately estimate the human body pose. The disadvantage of these methods lie in the fact that it is necessary to wear special suit, making it not suitable for real-life non-invasive applications. Additionally, sophisticated equipment is required, which is quite expensive. On the other hand, the usage of images or videos for tackling the HPE problem have gotten good advances during the last years. However, the images have some challenging to solve yet such as blurring, poor illumination among other.

This section starts presenting a review of state-of-the-art about human pose estimation applying classical methods and then, recent deep learning techniques using single-view and multi-view approaches are also reviewed.

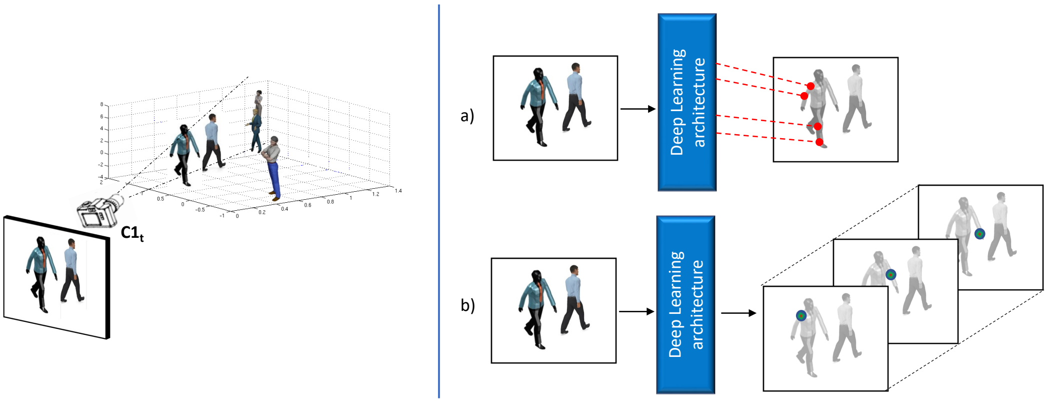


Figure 2.4: Human body pose estimation: a) Using regression-based approaches. b) Using heatmaps-based approaches.

### 2.2.1 Single-View Approach

In the past, the part-based modelling for facial structure estimation have been extended and used as a base in approaches like [30], to separately identify body parts. Each body part is configured to form deformable structures, which is connected between paired parts. Different deformable structures have been developed such as Adaptive Pictorial Structures, Multi-person Pictorial Structures, Poselet Conditioned Pictorial Structures [24, 80, 89] and others. In [29], the authors have proposed a new solution that requires that a set of relationships between parts form a tree structure considering that each pair of parts be of a particular form. The space of possible parts location is discretized into 50 buckets for each axis $(x, y)$. This solution minimizes an energy function for matching picture structure to images. Likewise, the authors in [25] have proposed a method to enhance pictorial structures, which use annotated images for learning relationships between the appearance of different body parts. A generic detector is used to estimate an approximate location and scale reference of the object, helping to generate better appearance models for body parts on new images. However, the results are limited since that local detector has some problem to correctly predict the body part due to complex poses, barely visible joints, even occluded joints among other.

With the advance of neural networks during last years, the research on HPE began to shift from classic approaches to CNNs. Two schemes are used to tackle HPE problem; one of them, is based on direct regression to estimate the body pose. The second one uses heatmap approaches. The latter predicts the probability of the joint occurring at each pixel (see Fig. 2.4).

Regarding regression-based approaches, the authors in [112] have tacked the HPE problem using CNNs. The approach contains stages which can refine the predicted pose for a next stage and thus, getting better results in the body joints estimation. Specifically, the images are cropped around the predicted joint and fed to the next stage, making that the next stage only see the region of interest (around predicted joint) to learn features and thus, improve accuracy. The model is based on AlexNet architecture with an extra final layer of k joints $(x, y)$ coordinates, it is included as output where k is the number of body joints. The authors in [111] have proposed a novel architecture that includes an efficient 'position refinement', which allows to estimate the joint offset within a small region of the image. Multiple resolutions banks are used in parallel to simultaneously capture the features in the image. For optimization process, the MSE is used between predicted heatmap and target heatmap, where the target is a 2D Gaussian of constant variance ($\sigma \approx 1.5$ pixels) centered at the ground truth $(x, y)$ joint location. Another approach is proposed in [7], where a self-correcting model is used to predict keypoint locations that are progressively refined instead of trying to directly predict the outputs in one step. The model requires as input an image and a representation of the previous output, refining the output of model through an iterative process. The model is based on GoogleNet architecture with $L_2$ regression as loss function.

In [76], the authors have proposed a novel architecture, called stacked Hourglass network. It consists of step of pooling and up-sampling layers, which captures global and local information at every scale, improving the recognition of the person's orientation and relations between joints. An intermediate supervision is applied to each hourglass in the different scale of the model, i.e., from high resolutions to low resolutions, and vice versa. Skip connections are used to preserve spatial information. A proposal using body bones instead of joints is presented in [103], since the bones are more stables. The geometric structure can be encoded more easily than joint as well as the geometric constraints. The authors in [130] have proposed a network structure with a few deconvolutional layers at the end of the model, using the ResNet [37] as backbone. Additionally, pose tracking follows a similar pipeline as in [32], adding the usage of the optical flow based pose propagation and similarity measurement. Similar as above, the MSE is used as a loss function between the predicted heatmap and ground truth.

The authors in [102] have proposed an architecture that contains high-resolution as first stage, then gradually high-to-low resolutions sub-networks are added, which connects everything in parallel, to maintain a reliable learning of features representation throughout the whole process of prediction. This approach performs repeated multiscale fusions to boost the high-resolution representations with the help of the low resolution representations of the same depth and similar level, and vice versa, which allow to have rich features for pose estimation. The process does not require intermediate supervision like in [76]. MSE loss function is used similar to the approaches mentioned above. Another approach is proposed in [17], where the authors have proposed an architecture to estimate expressive 3D humans from RGB images. However, the problem with these estimations (hands and faces) lies in the fact that represent few image pixels, making more challenging the problem. For tackling it, a bounding box is used to extract the body from the full resolution image. Additionally, the sub-networks of model are fed with face and hands, which are extracted from original images and thus, getting the final estimation for the face and hand parameters, thereby fine-tuning the final output.

A new method for 3D human pose estimation has been proposed in [36], where the static 3D scene structure is exploited from single view. Some constraints respect to the 3D pose estimation in the 3D scene are considered. One of them is that the pose is penalized if two objects in 3D space share the same space. Second one, a heuristic is used to determine the body parts that have more probability to contact the scene (for instance, the feet). It should be noted that the 3D scene and model of the human body should be accurate to get a good result. In [101], the authors have proposed a multi-view pose generator to predict multi-view 2D poses from a 2D pose in a single view. The 3D coordinates of the human body and the camera parameters are needed to generate a 2D pose corresponding to each camera. These generated 2D poses from each camera and 3D pose coordinates are used to train the proposed approach. In order to infer the 3D pose from estimated multi-view 2D poses, a graph convolutional network is used. Data augmentation is also proposed since that the dataset provides limited camera pose parameters. The MSE is used as loss function, including the bone length as constrain for learning process.

The authors in [5] have proposed an architecture to detect the 2D pose of multi-person from an image. The architecture contains two branches, which perform an iterative prediction, whereby the outputs of these branches are used as input to successive stages, considering intermediate supervision at each stage. The first branch is used to learn part locations and estimate a set of 2D vector fields of part affinities to encode the association between parts, while the second branch is used to learn the association between the joints and estimate a set of 2D confidence maps of each of them. $L_2$ metric is used as loss function, including a factor or binary mask to avoid penalizing predictions of missing joints during the training of the architecture. As mentioned above, the authors in [43] have also proposed a multi-person pose estimation approach. A body part detector is improved using as based modified ResNet architecture. This modification consists of stride shorter of sliding windows since this improves part localization, and although the large receptive field allows to predict more accurately a body part, this could have enough information about other localizations of other near body parts. Additionally, the proposal includes image-conditioned pairwise terms to assemble the proposal into a variable number of consistent body part configurations. In order to explore more efficiently the search space and reduce the run-time, an incremental optimization is used.

A normalizing flow-based method is proposed in [124]. This method intends to solve ambiguous inverse problems. A normalizing flow is a sequence of bijective transformations, which allows evaluation in both directions. Since it is a deterministic forward process (i.e., projection of the 3D poses to 2D) with multiple different inverse mappings, the human pose estimation is proposed as an ambiguous inverse problem from a single image. The model is optimized in both directions. A latent vector is produced through learning the 3D-to-2D mapping. For the inverse path, the 3D pose discriminator is used to penalize unfeasible poses. In order to model occlusions and uncertain detections, the uncertainty information from the predicted heatmaps of the 2D detector is used. An unsupervised approach for 3D human pose estimation from its 2D counterpart is proposed in [135]. The approach is divided into two sub-tasks. The first is to optimize the 2D input poses via a scale estimation module, and finally mapping optimized 2D-to-3D pose via lifting module. Additionally, two temporal constraints are proposed to tackle the scale and pose ambiguity problems. The temporal scale considers a bone constraint to optimize the scale of 2D pose at the frame level, while temporal pose ambiguity considers a geometric random rotation scheme to produce other view of 3D trajectories and construct multi-view information.

Although the CNNs are still used for the different computer vision tasks with appealing results, the transformer networks, specifically self-attention modules, just began to slowly creep into several computer vision tasks, being used as complement to CNN architectures or even completely replacing them. On this address, the authors in [63] have proposed to use the stacked Hourglass network, including it a polarized self-attention mechanism. This mechanism is added before the second convolution of the basic residual block, before the max pool down-sampling and the nearest neighbor up-sampling for each stage of Hourglass module. For maintaining a high feature resolution, is necessary to use the space and channel of the self-attention mechanism. With this, the accuracy of the model's positioning of the body key joints is improved. In [128] the authors have introduced a self-attention mechanism to simulate a nonlocal relationship between feature maps and combines long-term distance information into original feature maps. This generates an attention mask to reweight original features, which force to the model focus more on nonlocal information, making that the model can significantly increase performance and efficiency. The ResNet architecture is used as backbone to extract features from input images, and later to use the self-attention mechanism to get long-range dependency between all body joints. Finally, the results are sent to upsampling block to regress obtained feature maps to a higher resolution. Another approach is proposed in [122], where an attention refined network is used to enhance multi-scale feature fusion for human pose estimation. In order to reinforce important features of input image, channel and spatial attention mechanisms are considered, including a self-attention strategy that help to find long-range keypoints dependencies. Additionally, the model uses a focus loss where only 'hard' keypoints are used. These keypoints are selected considering the training loss during learning process and only backpropagates the gradients from the selected keypoints. The authors in [64] have proposed a human pose estimation framework, which is based on regression method. The pose estimation task is formulated as a sequence prediction problem, which can effectively be solved by transformers. The model is able to attend important features to the target keypoints using attention mechanism in transformers. The proposal has three main components: the first component corresponds to standard CNN backbone to extract multi-level feature representations, following of an encoder to capture and fuse these multi-level features, and finally, a coarse-to-fine decoder to generate a sequence of keypoint coordinates.

Unlike the previous approaches, the authors in [137] have presented a purely transformer-based approach for human pose estimation in videos without using CNN architectures. A spatial-temporal transformer structure is designed to model relations between body joints for each frame as well as the temporal correlations across frames. A sequence of detected 2D poses is used as input to the spatial self-attention layers to generate a latent feature representation for each frame and then, the global dependencies between each spatial feature representation are analyzed by temporal transformer module to obtain as output an accurate 3D human pose of the center frame. Other authors in [116] presented a Multi-level Attention Encoder-Decoder

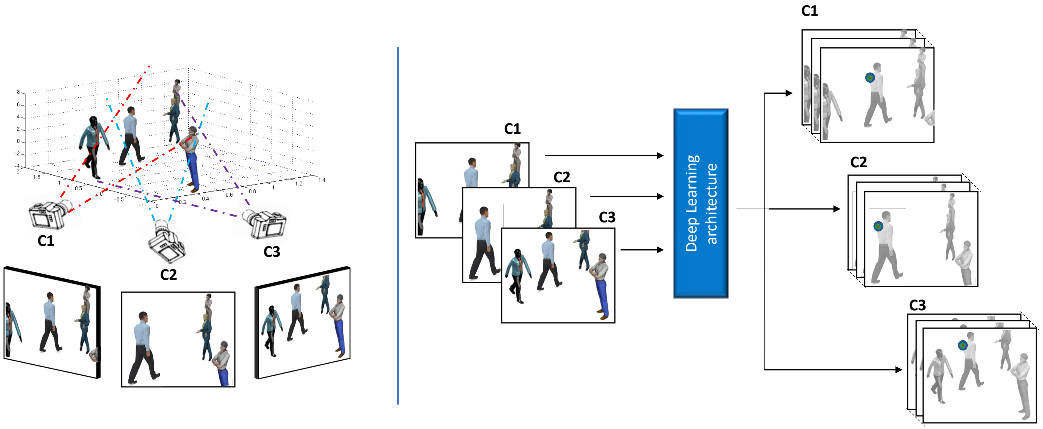


Figure 2.5: Human body pose estimation from multi-view approaches using Deep Learning architectures.

Network, which includes a Spatial-Temporal Encoder and a Kinematic Topology Decoder to model multi-level attentions in a unified framework. A CNN backbone is used to extract the basic feature for each frame, which are send as input to Spatial-Temporal Encoder. This encoder is designed as a series of cascaded blocks on Multi-Head self-attention, where two parallel branches are used for each block to learn spatial and temporal attention. Additionally, Kinematic Decoder models the joint level attention where each joint is assigned a unique linear regressor to get its pose and shape parameters. These predicted parameters, including camera parameters are utilized by parametric 3D human body model to calculate 3D joints and their 2D projection.

### 2.2.2 Multi-View Approach

Although the proposals for human pose estimation from one-view have shown appealing results, the problem of capturing the complex human body pose, either for self-occlusion of body parts or by occlusions from objects in the scene, has not yet been completely solved. An alternative to overcome the occlusions problems from monocular vision systems could be considered multi-view approaches. Basically, the human body is captured from different cameras and points of view at the same time, i.e., the occluded joints from one view could be observed from other views and thus, take advantage of redundant information (see Fig. 2.5). These multi-view approaches have been already used to overcome the region occlusion problem in certain tasks such as 3D-reconstruction, autonomous driving, object detection among others (e.g., [39, 91, 108, 131]). However, few works have tackled the occlusion body joints problems by means of multi-view approaches for human pose estimation. One of them is presented in [85], where the architecture predicts the same pose in all view, including a consistency constraint to predict accurate poses. Additionally, a small set of labeled images is used into supervised loss, which uses a regularization term that penalizes drift from initial predictions. A normalized pose distance is used to evaluate all losses involving poses. The intrinsic parameters are necessary for the proposed approach. However, the camera rotation is estimated using the subjects and their estimated pose as calibration target.

An unsupervised approach for 3D human pose estimation is proposed in [84], where the authors present an architecture to learn a geometry-aware body representation from multi-view images. For this end, the model required images of the same person, which should be captured from multiple views, to learn a latent representation. Since the 2D or 3D annotations are not required, the encoder-decoder is trained to predict an image seen from one point of view from other image captured from a different one. Additionally, the proposed approach considers some assumptions such as: to separate appearance from geometry, two frames of the same subject at different times are required, and that the person's appearance does not change drastically between both frames. The cameras of multi-view systems should be synchronized and calibrated before capturing the sequence of images. This could be integrated into semi-supervised approaches, and thus, reduce the required amount of supervised approaches. In [77], the authors have presented a deep learning-based method to estimate the 3D human pose through cascade system. The 2D joints are obtained for each available view. Each 2D estimated joint is analyzed qualitatively to decide either it is considered or reject into least square optimization to reconstruct the 3D coordinates of each point. The spatial and temporal information are combined using LSTM network to improve the 3D pose estimation. Specifically, the LSTM network receives as input a temporal sequence of 3D poses in successive time steps to produce a final sequence of 3D human pose.

Similar to the previous proposal, the authors in [81] present an approach to recover absolute 3D human pose from multi-view systems but incorporating multi-view geometric priors. The 2D poses from multi-view are obtained using a CNN based approach. The 2D pose heatmaps for each view are fused to find the corresponding locations between different views and thus, the heatmaps of one view benefits other views. Then a recursive pictorial structure is used to recover the 3D human pose from the estimated multi-view 2D pose heatmaps. The Joint Detection Rate is used as a measure to estimate the 2D pose distance between prediction and ground truth, while Mean Per Joint Position Error is used to determine the 3D pose estimation accuracy. Two-stage fully network to estimate 3D human pose is proposed in [40], where the authors use the multi-view images for the first stage to preserve data through multi-channel volume and 3D soft-argmax as activation layer. For the second stage, an Inertial Measurement Unit (IMU) layer is introduced into refinement stage to fuse the IMU and multi-view image data. In details, the Hourglass network is used as backbone, which takes as input the multi-channel volume while the output is a 3D voxel heatmaps of each joint. After, in the refinement stage, the IMU data and estimated joint from the last stage are fused to produce a set of refined voxel heatmaps and thus, obtain the 3D human pose estimation. In [83], the authors present a lightweight solution to recover 3D human pose from multi-view images, which are captured with spatially calibrated and temporally synchronized cameras. The ResNet-152 architecture is used as backbone for the encoder a set of multi-view images, each captured from a camera with known projection matrix. Additionally, the model exploits 3D geometry in latent space considering information about different views. The learned representations are conditioned on camera projection operators to produce accurate per-view 2D detection, while could be lifted to 3D structure using a Direct Linear Transform layer, being more efficient on GPU architectures than triangulation-based methods.

In [117], the authors have proposed a self-supervised approach, which does not require a labeled multi-view data for 3D pose estimation and works with uncalibrated cameras. A pre-trained 2D pose detector is used to estimate the 2D joint from images. At least two temporally synchronized cameras are required to capture the person of interest from different views. The predicted joints from each camera are lifted to a 3D pose, which are represented in a learned canonical coordinate system, including the cameras orientation prediction. Combining the predicted 3D pose from a first camera with the predicted camera orientation from a second camera, results in a rotated pose in the second camera coordinate system, which is leveraged into of a reprojection loss. The authors in [114] have proposed a model that only requires 2D keypoints data for training, including joint location uncertainty due to occlusion across multi-views. The proposed model predicts 3D human pose and relative camera poses from multi-view systems, for the latter, the human body joints are used as source of information for camera calibration and thus, to avoid consequent erroneous 3D pose predictions. An initial guess about the body pose and the camera setup are obtained using an off-the-shelf weakly-supervised 3D network. Additionally, an off-the-shelf 2D pose estimation network is used to generate 2D joint location probability heatmaps, which are iteratively refined by meta-optimizer.

On the other hand, like in the single-view approaches, transformer networks are being used for the human pose estimation from multi-view schemes, where these self-attention modules have showed appealing results. On this direction, the authors in [97] have proposed a unified framework to handle varying view numbers and video length without the need of camera calibration. Basically, a pre-trained 2D pose detector is used as backbone to estimate the 2D pose from each image and then, the predicted joints and its confidences are encoded into feature embedding for further 3D pose inference. As second step, the features of whole views are fused with a relative-attention block, which adaptively measures the implicit relationship between each pair of views and reconstructs the features. The reconstructed features are aggregated into a temporal fusing transformer to predict 3D pose via a transformer. In [121], Multi-view Pose transformer is presented to estimate multi-person 3D poses from multi-view images. A 2D pose detector proposed by [130] is used as backbone to obtain high-resolution image features from multi-view inputs, then the model represents skeleton joints as learnable query embeddings, reasoning and attending over the multi-view information from the input images. A geometry-guide projective attention mechanism is used to fuse the multi-view information, instead of applying full attention to densely aggregate features across spaces and views. The camera rays are encoded into the multi-view feature representations using a novel RayConv operation and thus, integrate multi-view positional information into the projective attention.

Similar to the approach presented above, the authors in [81] have proposed a transformer framework for multi-view 3D pose estimation, and thus, improve the individual 2D predictors using information from different views. The initial part of the ResNet-50 is used as backbone to capture the low features. The model fuses the captured features from both current views and neighboring views, including an encoding 3D positional information, which considers the concept of epipolar field into the transformer model. The 3D positional information is guided by epipolar field, which provides a way of encoding correspondences between pixels of different views.

## 2.3 Benchmark

This section discusses the needs of benchmark datasets for both the camera and human pose estimations. During the last years, several datasets have been created to tackle these problems. Some of them are described below.

### 2.3.1 Camera Pose Estimation

In order to tackle the camera pose estimation problem, some datasets are used for the training and evaluation processes of the different architectures, whether to solve it from a single-view system approach or multi-view system approach. The main datasets are shown below.

**7-Scenes Dataset**

*The 7-Scenes* dataset [96] is a collection of tracked RGB-D camera frames, which can be used to evaluate methods for solving problems such as tracking, mapping and relocalization. These scenes are Chess, Fire, Heads, Office, Pumpkin, Red Kitchen and Stairs (see Fig. 2.6). A handheld Kinect RGB-D camera at 640x480 resolution is used to record all scenes. The ground truth camera tracks and dense 3D model are obtained from the KinectFusion system, considering default intrinsics parameters for the depth camera. The homogeneous 4x4 matrix (rotation and translation) is used as label for the training and evaluation process. For more details, see Table 2.1.

Table 2.1: Details about 7-Scenes dataset. All scenes are available.

| Scene | Spatial Extent (m) | Number of Images |
|---|---|---|
| Chess | 3×2×1 | 6000 |
| Fire | 2.5×1×1 | 4000 |
| Heads | 2×0.5×1 | 2000 |
| Office | 2.5×2×1.5 | 1000 |
| Pumpkin | 2.5×2×1 | 6000 |
| Red Kitchen | 4×3×1.5 | 12000 |
| Stairs | 2.5×2×1.5 | 3000 |

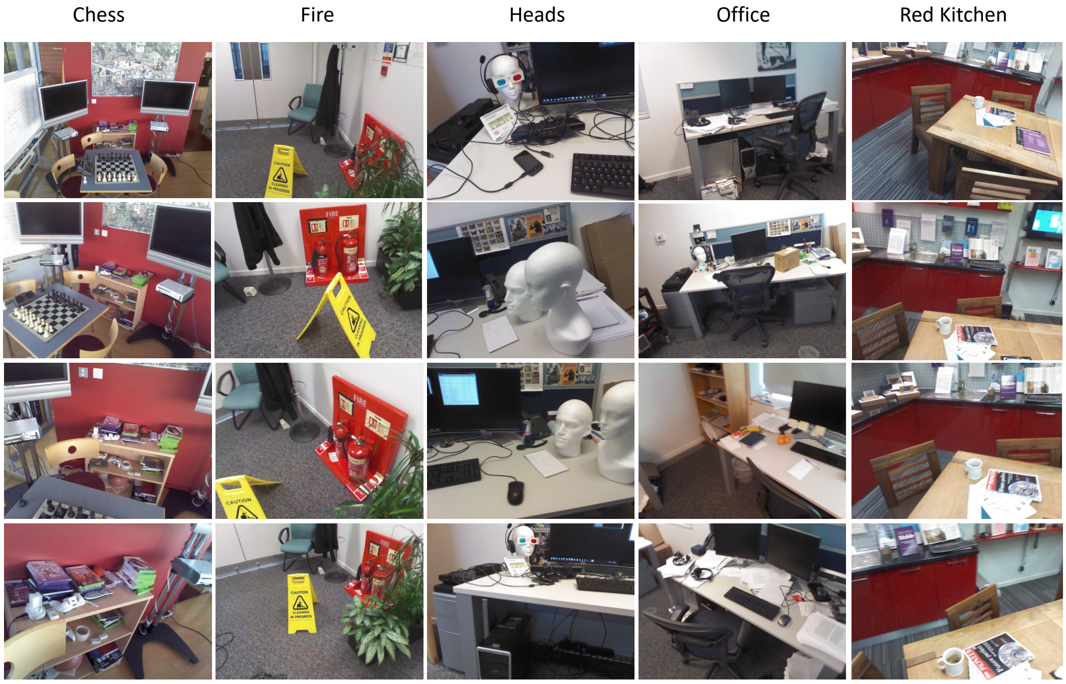


Figure 2.6: 7-Scenes dataset [96].

### Cambridge Landmarks Dataset

*Cambridge Landmarks* dataset [54] consists of five outdoor scenes referred to King's College, Shop Facade, Old Hospital, St Mary's Church and Street (see Fig. 2.7). A Google LG Nexus 5 smartphone is used by a pedestrian to take high definition video around each scene. The images are captured in a large scale outdoor urban scenario. Additionally, different lighting and weather conditions are used to collect the images. All images have a resolution of 1920x1080 pixels. The camera pose labels are generated using structure from motion techniques. The generated text-file contains a 3D-vector (x, y, z) and a 4D-vector (Quaternion), which are used as the translation and rotation between pairs of images respectively. For more details, see Table 2.2.

### DTU Robot image Dataset

*DTU Robot image* dataset [48] is aimed at multiple-view stereo evaluation. To get the images dataset, an industrial robot arm is mounted with a structured light scanner, which allows for structured light scans corresponding to each image in the dataset. The images are captured with a resolution of 1600×1200 pixels by one of the cameras in the structured light scanner from 49 or 64 different camera positions. A set of 124 different scenes are considered, including different lighting conditions from directional to diffuse. The scenes also include a

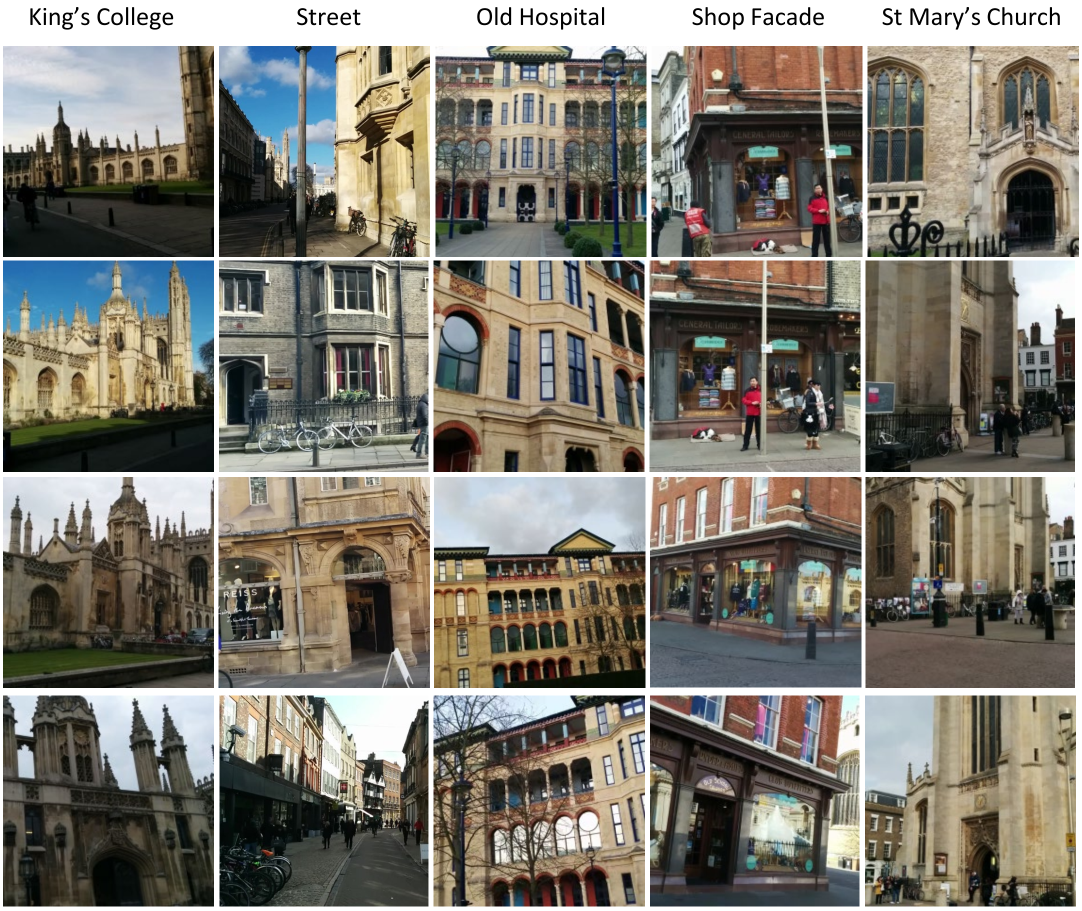


Figure 2.7: Cambridge Landmarks dataset [54].

Table 2.2: Details about Cambridge Landmarks dataset. All scenes are available.

| Scene | Spatial Extent (m) | Number of Images |
|---|---|---|
| King's College | 140×40 | 1563 |
| Street | 500×100 | 5938 |
| Old Hospital | 50×40 | 1077 |
| Shop Facade | 35×25 | 334 |
| St Mary's Church | 80×60 | 2017 |

wide range of objects to span the multiple-view stereo problem with very similar objects to the real-world, e.g., model houses. Some scene are shown in Fig. ??. The Matlab calibration toolbox is used to find the camera parameters. The dataset provides text-files with camera matrix for each position.

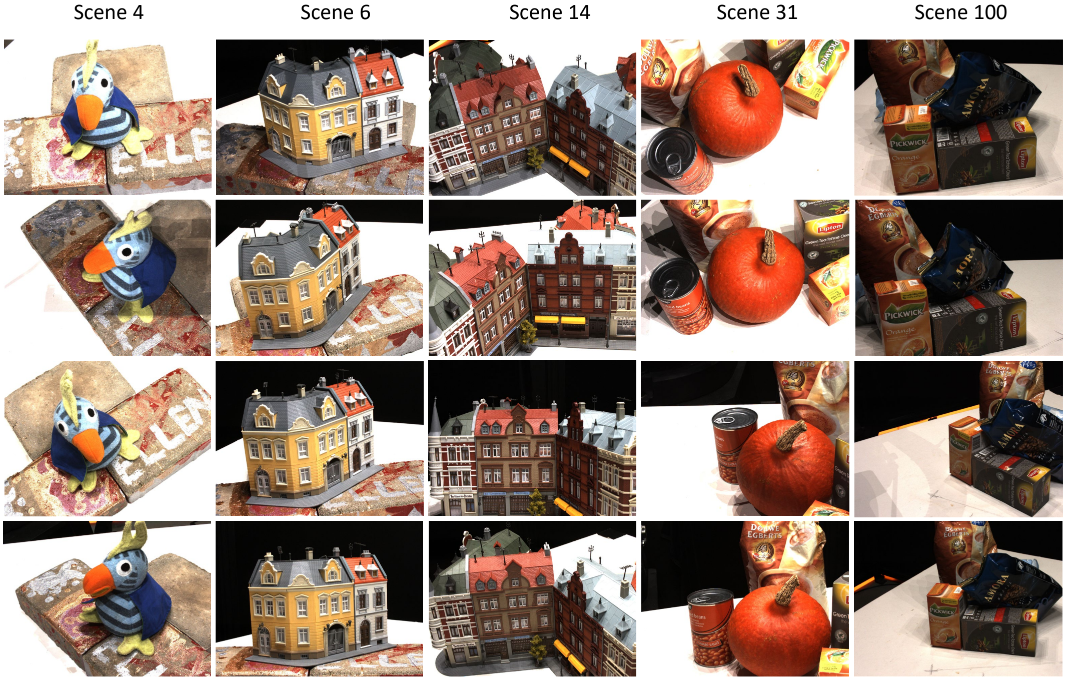


Figure 2.8: DTU Robot dataset [48].

### 2.3.2 Human Pose Estimation

A brief summary of the main datasets used for the human pose estimation problem is presented below. Some of these datasets could be considered to tackle this problem from a multi-view approach.

#### Leeds Sports Pose Dataset

*Leeds Sports Pose* dataset [49] contains 2000 images with human body pose annotations, mostly of sport activities gathered from Flickr (see Fig. 2.9). A file-text with joints annotations is provided to be used as ground truth. These annotations are represented by a 2D-vector (x, y), including a value that indicates the visibility of each joint. A total of 14 joint locations are available. Left and right joints are consistently labelled from a person-centric viewpoint. Most of the image resolutions have approximately 1024×700 pixels, which depend on the sport activity.

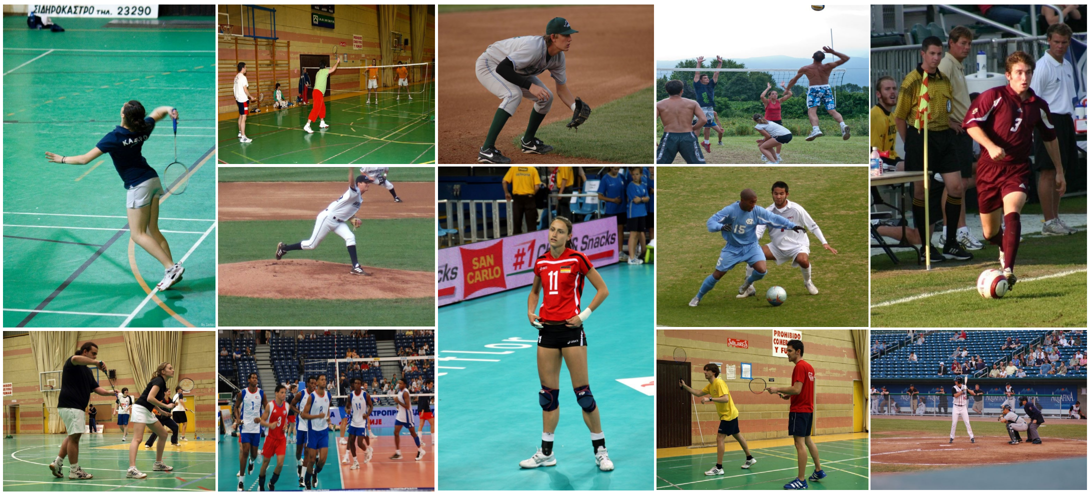


Figure 2.9: Leeds Sports Pose dataset [49].

#### COCO Dataset

*Common Objects in Context* knows as *COCO* [60], is a large-scale image dataset that contains 328k images in which more than 200k are labelled images. They could be used to tackle object detection and segmentation problems (see Fig. 2.10). For this, 80 object categories (e.g., person, bicycle, elephant) and 91 stuff categories (e.g., grass, sky, road) are available. Most of the images have a resolution of 640×480 pixels approximately. The dataset has 250k images with persons where keypoints are annotated (x,y) coordinate of 17 possible keypoints, such as shoulder, head, knee, ankle, etc. Additionally, densePose annotations are provided with more than 39k images and 56k person instances labeled where each labeled person is annotated with an instance ID and mapped to a template 3D model.

#### MPII Human Pose Dataset

*MPII Human Pose Dataset* [2] is used for evaluation of articulated 2D human pose estimation for multi-person. It contains around 25K images, which include over 40k persons with annotated body joints, which are manually annotated with up to 16 body joints in pixel coordinates (x, y), including a visibility value of joint. Each image is extracted from a Youtube video with no annotated frames, where 410 human activities are covered with corresponding labels (see Fig. 2.11). The images are systematically collected using an established taxonomy of every day human activities. The rough human position in the image is provided, including person scale with respect to 200 pixels of height.

Figure 2.10: COCO dataset [60].

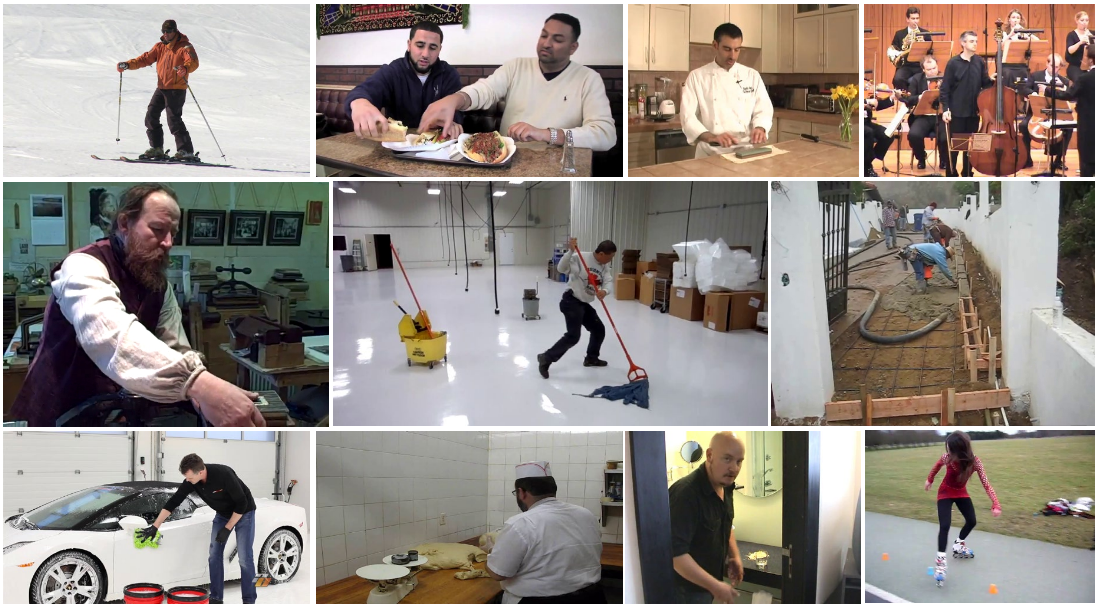


Figure 2.11: MPII Human Pose dataset [2].

### MPI-Inf-3DHP Dataset

*MPI-Inf-3DHP Dataset* [69] is captured in a multi-camera studio with ground truth from commercial marker-less motion capture, and contains approximately 1.3M images (see Fig. 2.12). The images have a resolution of 2048×2048 pixels. Eight actors are considered to perform eight different activities each one, ranging from walking and sitting to complex exercise poses and dynamic actions. No special suit and marker are required to record the different actions performed by the actors. A total of 14 cameras are strategically located to cover a wide range of viewpoints. In details, five cameras are mounted at chest height with a roughly 15° elevation, another five cameras are mounted higher and angled down 45°, three cameras have a top down view, and the last camera, is at knee height angled up. A text-file of camera calibration is also provided, including body joint annotations in each camera's coordinate system.

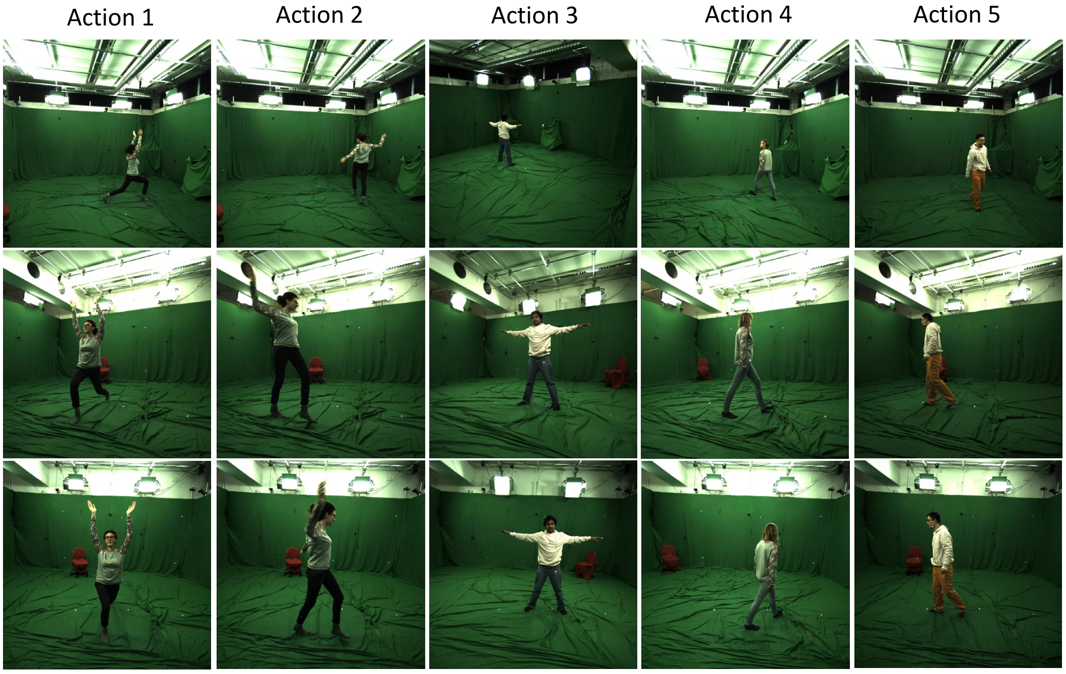


Figure 2.12: MPI-Inf-3DHP dataset [69].

### Total Capture Dataset

*Total Capture Dataset* [113] is designed for 3D human pose estimation from multi-camera system, and contains approximately 1.9M RGB images (see Fig. 2.13). The dataset has fully synchronised multi-view video, IMU and Vicon labelling for a large number of frames. The images have a resolution of 1920×1080 pixels. A text-file of camera calibration is also provided to obtain the 2D body joints. There are 13 sensors on key body parts such as head, upper/lower back, upper/lower arms, legs and feet to provide the IMU data, including orientation and acceleration denoted by quaternion and 3D-Vector (x, y, z) respectively. The different actions performed by subjects are captured into indoor environment where there are 4 male and 1 female subjects, which perform actions such as walking, acting and freestyle, into a volume measuring roughly 8x4m with 8 calibrated HD video at 60Hz. Respect to the acting and freestyle sequences, these are very challenging with actions such as yoga, giving directions, bending over and crawling, which are available in both the training and evaluation processes.

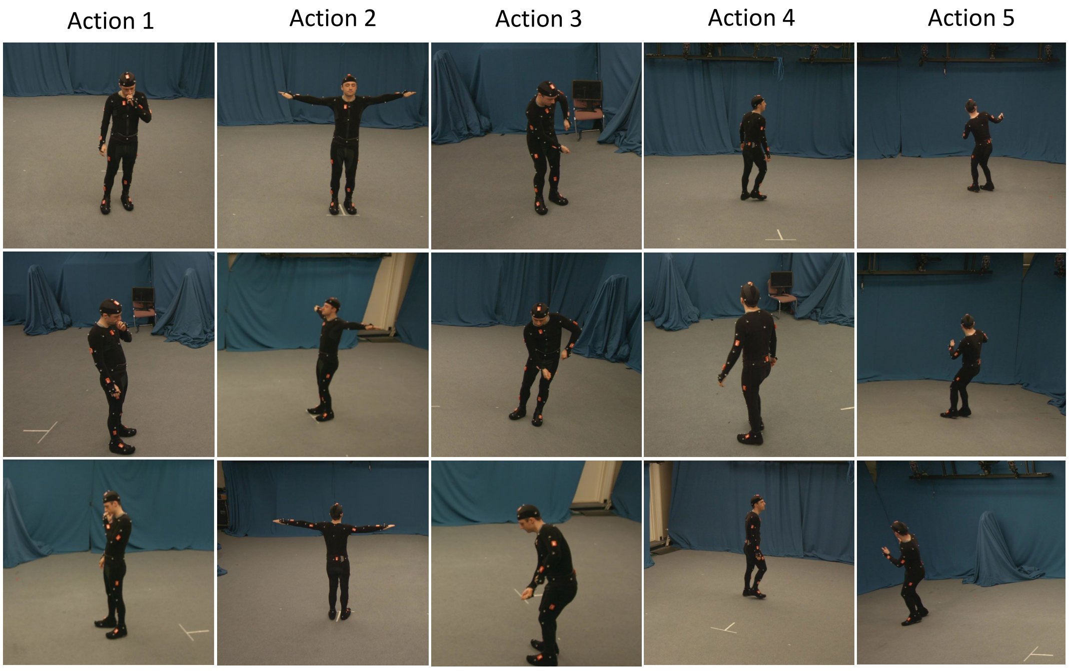


Figure 2.13: Total Capture dataset [113].

### CMU Panoptic Dataset

*CMU Panoptic* [50] is a large scale dataset that contains 1.5M of 3D skeletons to tackle the 3D body pose and 3D face landmark estimations. The dataset captures 86 subjects performing common tasks of daily living such as walking, jogging, sit-to-stand among other, including group interaction scenes (see Fig. 2.14). It can be used to solve one-person or multi-person body pose estimation problem. There are 19 body joints available as output of each image, which are represented by a 3D-vector (x, y, z) in the world coordinates system. The intrinsic parameters are available for each camera. The captured scene contains the following hardware setup: 480 VGA camera with a resolution of 640×480 pixels, which are synchronized among themselves using a hardware clock; 31 HD cameras of 1920×1080 pixels resolution, which are also synchronized with VGA camera using a hardware clock; 10 Kinect II sensors with a RGB resolution of 1920×1080 pixels, and a depth resolution of 512×424 pixels, which are timing aligned among themselves and other sensors; and 5 Digital Light Processing projectors synchronized with HD cameras.

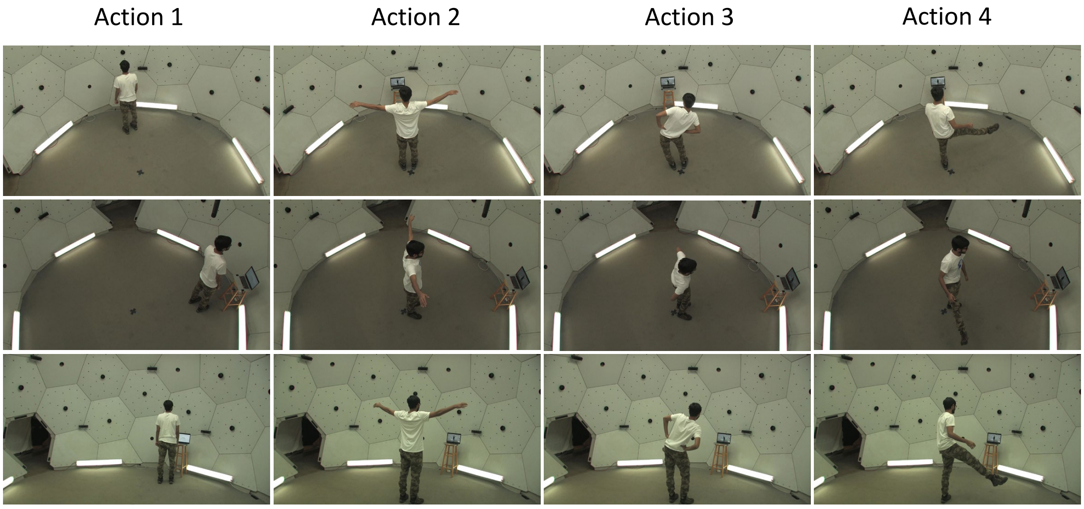


Figure 2.14: CMU Panoptic dataset [50].

### Human3.6M Dataset

*Human3.6M* [44] is a large scale dataset that contains 3.6 million of different human pose collected from 15 sensors (4 digital video cameras, 1 time-of-flight sensor, 10 motion cameras), which are synchronized and calibrated. The high resolution digital cameras are used to acquire video data at 50Hz with a resolution of 1000×1000 pixels. They are on same clock and trigger as the motion capture system, which ensures perfect synchronization between the video and pose data. The area of scene has a dimension of about 6×5m, and within this area a 4×3m of effective capture space is defined, where subjects are fully visible in all video cameras. There are 11 professional actors (6 male and 5 female), which wore their own regular clothing, as opposed to special motion capture costumes, to maintain as much realism as possible. The dataset is organized into 15 training motions such as directions, eating, greeting, taking photo, posing, smoking, walking among other, including information about the 3D positions (x, y, z) of each joint of the human body in world coordinates system and kinematic representation (see Fig. 2.15). On regard the late, it considers the relative joint angles between limbs and is more convenient because it is invariant to both scale and body proportions. The 2D joint positions are also provided considering the camera parameters.

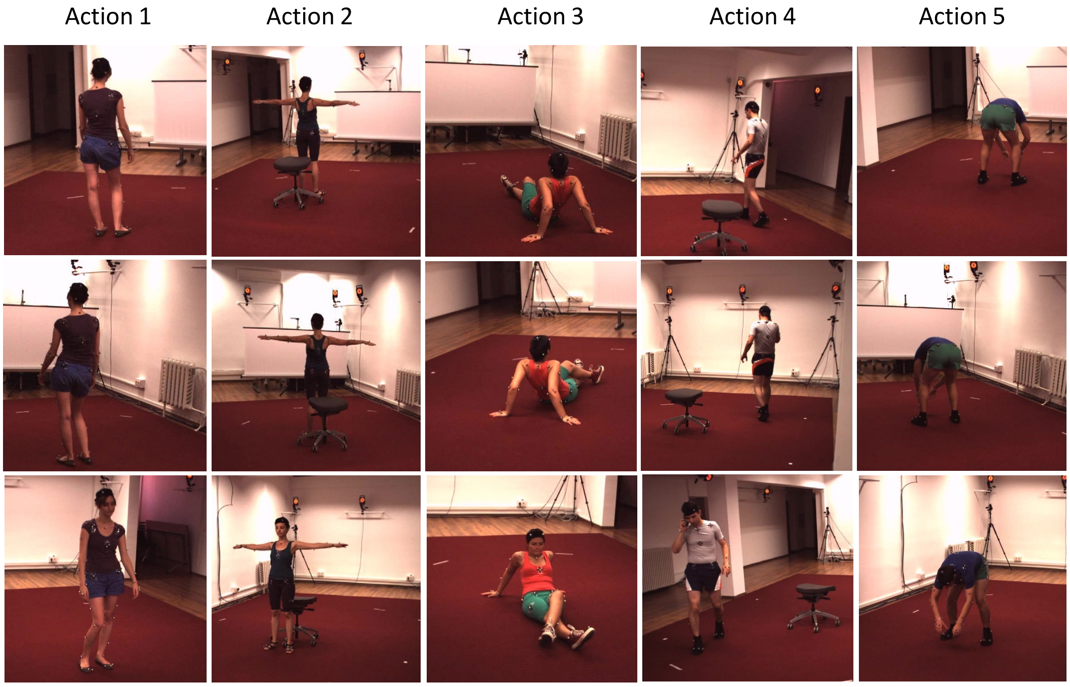


Figure 2.15: Human3.6M dataset [44].

# 3 Camera Pose Estimation

This chapter focuses on how to tackle the camera pose estimation problem in multi-view environments, mainly when large datasets are scarce for algorithm training, being one of the research questions proposes in this thesis. Some architectures are proposed to tackle this problem using RGB images from multi-view environments, including a Domain Adaptation strategy. The architectures proposed in this chapter are based on a Siamese CNN model that allows to find a relationship between pairs of images, which have been obtained from a multi-view system. A Domain Adaptation strategy is considered to facilitate the training process. Additionally, it is used to evaluate the need for similarity between real and synthetic image datasets to improve the camera pose estimation.

## 3.1 Introduction

The calibration process is an important step to get the camera parameters, which is required in most of the computer vision tasks. This process is usually performed by using special calibration patterns. Some proposals to tackle it in an automatic way have been developing during last decade, using just the context of the scene captured in the image without considering any special calibration pattern. Different difficulties in the context of the images such as illumination, low resolution and few image features, make the accuracy of camera calibration decreases when the classic algorithms are used. Hence, different CNN architectures have been proposed during the last years to overcome these problems due to the power to extract features images, showing better results than classical approaches.

Nowadays, the camera calibrations, specifically the relative camera pose, could be tackled from a single or a multi-view environment. The first case, single-view environment, has a main limitation since the scene is captured by one camera from a same position, doing that some important features could be occluded due to the camera's position in the world coordinate system respect to the captured objects. For the second case, multi-view environment, the problem of occluded features could be solved since the scene is captured from multi-views, i.e., different points of views are considered to capture the scene at the same time. However, it is necessary to ensure the overlap between the acquired images.

The contributions of this chapter focus on answering one of the research questions, which seeks to solve the camera pose estimation problem in multi-view environments, mainly when there is a lack of large datasets. These contributions are as follows:

- Develop a Siamese network architecture to obtain the relative camera pose between pairs of images from multi-view environments.
- Generate different synthetic datasets containing pairs of images from different scenarios.
- Develop a domain adaptation strategy to train a network for tackling the relative camera pose estimation and the lack of large datasets problems.
- Show the importance on the similarity between the real and synthetic images; in other words geometric similarity (i.e., 3D models in real and virtual scenarios) and point of view similarity (i.e., distance between camera and objects in the scene and camera orientation).

This chapter results on the following publications

- **Jorge L. Charco**, Boris X. Vintimilla, and Angel D. Sappa. Deep learning based camera pose estimation in multi-view environment. In International Conference on Signal-Image Technology & Internet-Based Systems, pages 224–228, 2018.
- **Jorge L. Charco**, Angel D. Sappa., Boris X. Vintimilla., and Henry O. Velesaca. Transfer learning from synthetic data in the camera pose estimation problem. In International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, pages 498–505. INSTICC, SciTePress, 2020.
- **Jorge L. Charco**, Angel D. Sappa, Boris X. Vintimilla, and Henry O. Velesaca. Camera pose estimation in multi-view environments: From virtual scenarios to the real world. Image and Vision Computing, Vol.110:104182, 2021.

## 3.2 Network Architecture

Neural networks, also known as artificial neural networks (ANNs), are a class of artificial intelligence, which allows to reproduce intelligent decisions, as done by humans. The ANNs are inspired by structure of the human brain, allowing that computer programs recognize patterns by using mathematical models, which help to the learning process of neural networks. The perceptron is a neural network unit, which performs certain calculations taking in account input data, weights and threshold value, including an activation function to produce binary

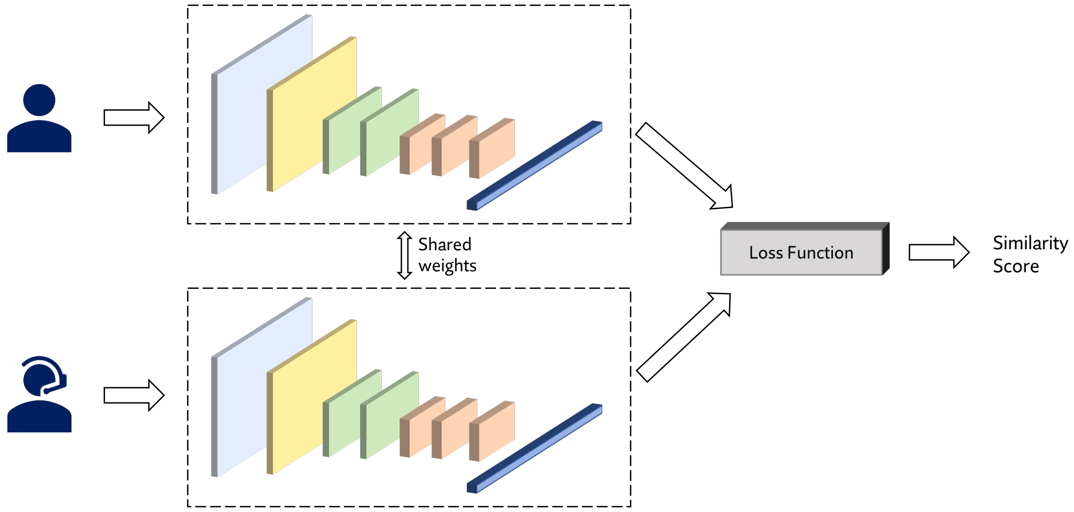


Figure 3.1: General Scheme of Siamese Neural Network [15].

output. Other type are the multi-layer perceptrons, which are comprised of node layers as input, hidden layers, which could be one or two layers, and an output layer. Each node of layers has a weight and threshold that allow to transform the input information to relevant information. Currently, there are other types of neural network architectures, which can be used to solve different task of real-world due to the power of modeling complex relationship among data, such as convolutional neural network, recurrent neural network, generative adversarial networks, including Siamese neural network (SNN).

A SNN [15], also known as Twin Neural Network, is an architecture that contains two or more identical sub-networks, which have the same configuration, i.e., the same parameters and weights (see Fig. 3.1). The parameters are updated during the training process in both sub-networks considering a similarity function. They are used to find the similar features between inputs vectors. Some common tasks such recognizing handwritten checks, automatic detection of face in camera images, and matching queries with indexed documents could use these architectures. Most well-known application using SNN architectures is face recognition, which is tackled as facial recognition problem to find a one person among a large number of other persons. However, other problems could be also tackled using Siamese CNN architectures such as object tracking and camera pose estimation. The latter, with a slightly different function in the output of the architecture.

### 3.2.1 Siamese model

In this thesis, a novel Siamese architecture has been proposed, which is referred to as RelPoseTL. It consists of two identical branches of a modified Resnet-50 architecture (see Fig. 3.2).

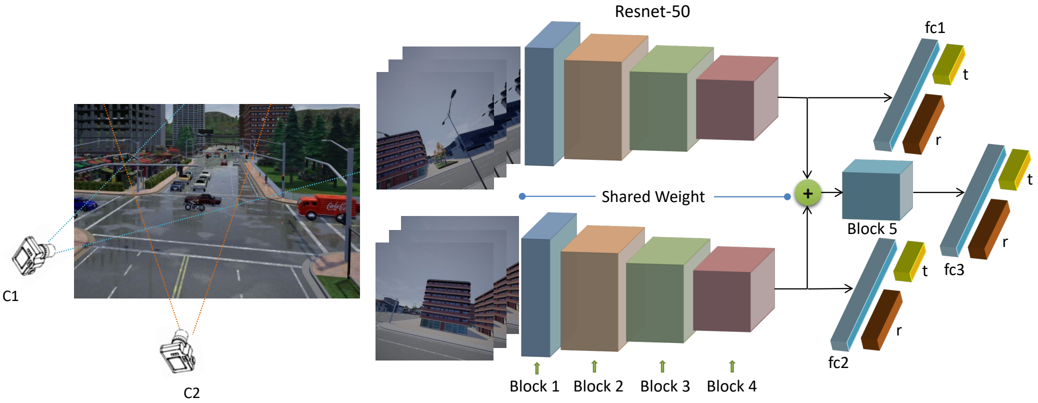


Figure 3.2: Siamese CNN architecture fed with pairs of images of the same scene captured from different points of views at the same time. The extrinsic camera parameter estimation is obtained by using the regression part, which contains three fully-connected layers.

The weights are shared up to the fourth residual block. Then, the output of each branch is concatenated to feed the fifth block of the used model. Multiple residual units are used, including convolutional layers, batch normalization, pooling, and identity blocks. The activation function of the Resnet-50 architecture is replaced by ELUs function in the proposed model, since according to [18], ELU helps to speed up convergence, avoiding the vanishing gradient. Three fully connected layers called *fc1*, *fc2* and *fc3* are added on the top of the proposed architecture. The two first fully connected layers (*fc1*, *fc2*) are fed with the output of each branch (i.e., after the fourth residual block). The third fully connected layer (*fc3*) is added after the fifth residual block. The **global pose** can be predicted for both cameras through the features extracted up to the fourth residual block, which are then used to feed the fully connected layers mentioned above. Each fully connected layer, i.e., *fc1* and *fc2*, has a dimension of 1024 followed by two regressors, which help to predict the global translation in 3D vector-format (*x,y,z*) and rotation in quaternion-format (*w,i,j,k*). In contrast to the approach mentioned above, the **relative camera pose** is obtained between both cameras by two regressors, which are fed from the output of the fully connected layer (*fc3*), which has a dimension of 1024 and is connected to the fifth block of the used model. These regressors are used to estimate the relative translation in a 3D vector (*x,y,z*) and rotation in a quaternion (*w,i,j,k*) between the given pair of images.

Relative camera pose is represented by the following equation: $\Delta p = [\hat{t}, \hat{r}]$, where the translation $\hat{t}$ is represented by a 3D vector (*x,y,z*), and the rotation $\hat{r}$ is represented by a 4D vector represented by a quaternion (*w,i,j,k*). As the images are captured at the same time on the same environment, it is reasonable to build one model that is able to predict at the same time the translation and rotation between the pair of images. Normally, the Euclidean distance is used to compute the error between the estimation and ground truth:

$$T_{Global}(I) = \left\| t - \widehat{t} \right\|_{\gamma}, \tag{3.1}$$

$$R_{Global}(I) = \left\| r - \frac{\hat{r}}{\|\hat{r}\|} \right\|_{\gamma}, \tag{3.2}$$

where the ground truth translation is denoted by $t$ and the predicted translation of used model is represented by $\hat{t}$. Likewise, $r$ is the ground truth rotation and $\hat{r}$ denotes the prediction of the quaternion values, which are normalized to a unit length by using $\frac{\hat{r}}{\|\hat{r}\|}$. $\gamma$ is $L_2$ Euclidean norm. Both components (i.e., translation and rotation) are trained together using the same loss function, including a factor $\beta$ to balance both components due to the difference in scale, similar to [54]. Hence, the proposed loss function is defined as:

$$Loss_{global}(I) = T_{Global} + \beta * R_{Global}, \tag{3.3}$$

since the $\beta$ parameter depends on several factors (e.g., scene, camera, scale, among others) finding the right value becomes a challenging task. Hence, in order to solve it, two learnable variables called $\hat{s}_x$ and $\hat{s}_y$ proposed by [52] are used. These variables act as weights that balance translation and rotation terms, generating a similar effect as $\beta$ parameter. The modified loss function, which uses the learnable variables, including the Eq. (3.1) and Eq. (3.2), is as follow:

$$Loss_{Global}(I) = (T_{Global} * exp(-\hat{s}_x) + \hat{s}_x) + (R_{Global} * exp(-\hat{s}_y) + \hat{s}_y), \tag{3.4}$$

The loss function mentioned above is useful to independently estimate the global pose for each pair of images using the corresponding branch of the proposed architecture. On the other hand, the output of the fifth block is needed to estimate the relative pose between both cameras, which is obtained as follow:

$$T_{Relative}(I) = \left\| t_{rel} - \hat{t}_{rel} \right\|_{\gamma}, \tag{3.5}$$

$$R_{Relative}(I) = \left\| r_{rel} - \frac{\hat{r}_{rel}}{\|\hat{r}_{rel}\|} \right\|_{\gamma}, \tag{3.6}$$

where $T_{Relative}$ and $R_{Relative}$ estimate the relative camera pose between the ground truth and the prediction obtained by the proposed architecture ($\hat{t}_{rel}$ and $\hat{r}_{rel}$). Similar to Eq. (3.2), $\hat{r}_{rel}$ is normalized. In order to obtain $t_{rel}$ and $r_{rel}$, the Eq. (3.7) and Eq. (3.8) are considered:

$$t_{rel} = t_{C1} - t_{C2}, \tag{3.7}$$

$$r_{rel} = r^{*}_{C2} * r_{C1}, \tag{3.8}$$

where $Ci$ corresponds to the pose parameters of the camera $(i)$, referred to as a global reference

Figure 3.3: Real-world images: *Shop Facade* and *Old Hospital* datasets respectively, used to evaluate the proposed model [54].

system. In order to obtain the relative translation between both cameras, the Eq. (3.7) is used, while to compute the relative rotation between both cameras, the Eq. (3.8) is considered. In order to compute the relative rotation it is necessary to use the conjugate quaternion [20] of $r_{C2}$, named as $r^*_{C2}$. Finally, the loss function used to obtain the relative pose considers the Eq. (3.5) and Eq. (3.6), and is defined as follows:

$$Loss_{Relative}(I) = (T_{Relative} * exp(-\widehat{s}_x) + \widehat{s}_x) + (R_{Relative} * exp(-\widehat{s}_y) + \widehat{s}_y). \tag{3.9}$$

where $exp(-\widehat{s}_i)$ corresponds to the exponential of trainable parameters, and thus to give valid values for variance according to [52].

Note that the global and relative camera pose parameters are obtained through two different equations. The first one, Eq. (3.4), predicts the global camera pose using the output from each branch of the trained model, while the second one, Eq. (3.9), predicts the relative camera pose through of the output of the fifth block of trained model (see Fig. 3.2). The final loss function for training the proposed architecture is defined as follows:

$$L = Loss_{Global} + Loss_{Relative}. \tag{3.10}$$

### 3.2.2 Experimental Results

One outdoor environment dataset to tackle the camera pose estimation problem called *Cambridge Landmarks* is used to conduct the experiments. The euclidean distance and angular errors are considered to evaluate the proposed model. This section will briefly describe used dataset, metrics and the obtained results.

### Dataset

Shop Facade and Old Hospital outdoor scenes, which become to *Cambridge Landamarks dataset* proposed by [54], are two of the five scenes used to evaluate the proposed model (see Fig. 3.3). These two scenes are captured using a Google LG Nexus 5 smartphone, which allow to take high definition video. Different lighting and weather conditions are considered to collect the images. The images have a resolution of 1920×1080 pixels. The camera pose labels are generated using structure from motion techniques (i.e., a file that contains 3D-vector for translation and 4D-vector for rotation between pairs of images).

### Metrics

In the camera pose estimation problem, angular error and Euclidean distance error are used to evaluate the performance of trained model. The first one, compute the rotation error between the obtained result (i.e., a 4-dimensional vector (quaternion)) and the ground truth values. The second one, is used to measure the distance error between the estimated translation (i.e., a 3-dimensional vector) and the corresponding ground truth.

### Training

All layers of modified Resnet-50 architecture were initialized up to the fourth residual block with the weights of Resnet-50 pretrained on ImageNet and the normal distribution initialization was used for the remaining layers. The network architecture was implemented with Keras and trained with NVIDIA Titan XP GPU and Intel Core I9 3.3GHz CPU. Adam optimizer is used to train the network with a learning rate of $10^{-4}$ and batch size of 32. The $\hat{s}_x$ and $\hat{s}_y$ are initialized with -3.0 and -6.5 in the experiments respectively. The network architecture was trained on Shop Facade and Old Hospital of *Cambridge Landmarks* dataset. As previous step, a pre-processing dataset was performed, which consisted of resizing all images to 224 pixels along the shorter side. The mean value was computed and subtracted from the images as final step of pre-processing dataset. For the training process, a random crop is used, which is a data augmentation technique to create a random subset of an original image. According to mentioned in [55, 107, 138], this technique helps to the learning process since the models could be able to generalize better the knowledge learned to similar scenarios. Considering this, random crops of 224×224 pixels have been computed on a set of 5900 pairs of images from *Old Hospital* dataset. The same process has been performed at 1300 pairs of images from *Shop Facade* dataset. Both datasets were used to train the network until 500 epochs, which approximately took 7 hours and 3 hours respectively.

The pre-processing mentioned above has been also performed during the evaluation process. In this phase, a set of 2100 pairs of images from *Old Hospital* dataset and a set of 240 pairs of images from *Shop Facade* dataset have been considered. On the contrary to the training process, a central crop is performed for all images instead of a random crop.

Table 3.1: Comparison of average median relative camera pose errors (extrinsic parameters) of RelPoseTL with respect to Pose-MV on *Shop Facade* and *Old Hospital* datasets.

| Scene / Models | Pose-MV [12] | RelPoseTL |
|---|---|---|
| Shop Facade | 1.126m, 6.021° | **1,002m, 3.655°** |
| Old Hospital | 5.849m, 7.546° | **3.792m, 2.721°** |
| Average | 3.487m, 6.783° | **2.397m, 3.188°** |

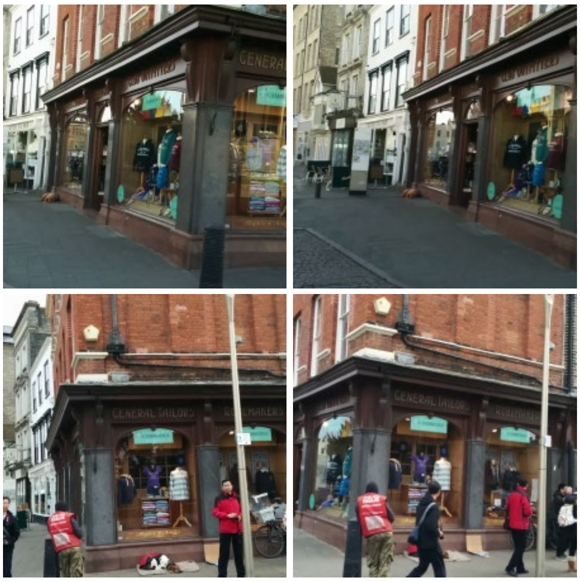


Figure 3.4: Challenging scenarios, pairs of images from points of view with large relative rotation and translation, as well as moving objects [54].

### Results and Comparisons

Experimental results obtained with the proposed network are presented. These results are compared with a state-of-the-art CNN-based method Pose-MV on *Shop Facade* and *Old Hospital* datasets. The Pose-MV is an architecture we have previously proposed in [12] to tackle the camera pose estimation from multi-environments, which is a Siamese CNN architecture implemented using as baseline a modified AlexNet architecture.

Average median error on rotation and translation for both datasets are depicted on Table 3.1. The proposed approach obtains more accurate results on both translation and rotation in both datasets. The average median translation error obtained by RelPoseTL improves the results of Pose-MV in about 32% (both approaches trained with the same real data). With respect to average median rotation error, RelPoseTL improves by 53% the results obtained with Pose-MV. Large errors are generated by challenging scenarios such as those presented in Fig. 3.4.

## 3.3 From virtual environments to the real world

One problem of CNN architectures is that they require training data of sufficient quantity and certain quality (e.g., rich in texture, different scenarios) to get accurate results. However, in many areas, the training data are not sufficient or difficult to acquire. Recent advances in computer graphics have allowed to create realistic virtual environments to simulate real-world environments with high quality, being an alternative to tackle this issue. The benefits and usefulness of virtual environments to train artificial neural network, specially the CNN architectures, have been demonstrated in many areas such as healthcare [68], human interaction [75] among others.

Some software tools to create virtual environments are available. These include Unity3D [132], Unreal Engine [82], Blender [98], CARLA [23], just to mention a few (see Fig. 3.5). An advantage of using virtual environments lies on the fact that on the one hand, it is possible to generate almost an unlimited set of synthetic images; on the other hand, a large variability, containing different conditions, actors and multiples cameras, may be considered; for instance, in the case of pedestrian detection for video surveillance or driving assistance, scenes with different weather conditions, illuminations, pedestrian's shapes, clothes, etc. can be considered during the dataset generation (i.e., synthetic image acquisition). An additional advantage when virtual environments are considered lies on the fact that ground truth are automatically obtained, reducing human error when the datasets are manually annotated.

### 3.3.1 Virtual Environments

In order to tackle the limitation of datasets for the camera pose estimation problem, CARLA simulator is used. It is an open-source software that has an editor that allows modifying the existing virtual world; this editor integrates both CARLA simulator and Unreal Engine, which is a video game engine on which this simulator is based on. In addition, the open source software tool called OpenMVG [73] is used to guarantee a minimum overlap between a given pair of synthetic images.

The different synthetic image datasets are obtained by the CARLA simulator as indicated below. First, the path to be followed by a vehicle configured for this task is defined. Two virtual cameras are considered; they are placed at two different positions on the vehicle. Second, the virtual cameras are configured to generate images with the following attributes: width=2560, height=2560 and FOV=100. Both cameras start the trajectories with predetermined values in their position and orientation (see Table 3.2); then, these cameras perform the acquisition of images at the same time, which generates a file with the positions (x, y, z) and rotation (pitch, yaw and roll) with respect to the world references system. The process of acquiring, approximately 3000 synthetic images, from both synchronized cameras, for each dataset, takes

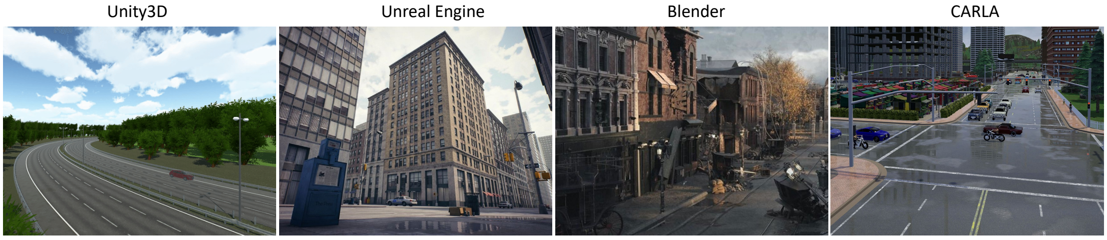


Figure 3.5: Synthetic images generated from software development tools [23, 82, 98, 132].

Table 3.2: The initial position and orientation of the two cameras with respect to the global reference system for each virtual scene.

| Scene | x | y | z | pitch | yaw | roll |
|---|---|---|---|---|---|---|
| Dataset 1 | -52.0 | -142.4 | 10.5 | -17.0 | 90.0 | 0.0 |
| Dataset 2 | -52.0 | -148.4 | 5.5 | 7.0 | 90.0 | 0.0 |
| Dataset 3 | -25.0 | 133.8 | 3.0 | 15.0 | 90.0 | 0.0 |
| Dataset 4 | -29.0 | 137.8 | 10.5 | -17.0 | 90.0 | 0.0 |
| Dataset 5 | 71.0 | 238.5 | 4.0 | -17.0 | 270.0 | 0.0 |
| Dataset 6 | 71.0 | 238.5 | 11.5 | -17.0 | 270.0 | 0.0 |

approximately three hours. Third, the OpenMVG software is used to verify the percentage of overlap between the pairs of generated synthetic images. The pairs of images that meet the overlap restriction of at least 60% are used to generate a file containing the cameras' relative pose information. The configuration options of OpenMVG software have been set as presented in Table 3.3. The ground truth for the global camera pose has been obtained from CARLA simulator, while the relative camera pose is computed from Eq. (3.7) and Eq. (3.8).

Table 3.3: Options available for OpenMVG script.

| **Order** | **Option** | **Used** |
|---|---|---|
| 1 | Intrinsics analysis | Yes |
| 2 | Compute features | Yes |
| 3 | Compute matches | Yes |
| 4 | Do Incremental/Sequential reconstruction | No |
| 5 | Colorize structure | No |
| 6 | Structure from known poses (robust triangulation) | No |

The *Cambridge Landamarks* dataset, which has different real-world scenarios [54], specifically, King's College and Old Hospital scenarios have been used as reference to generate

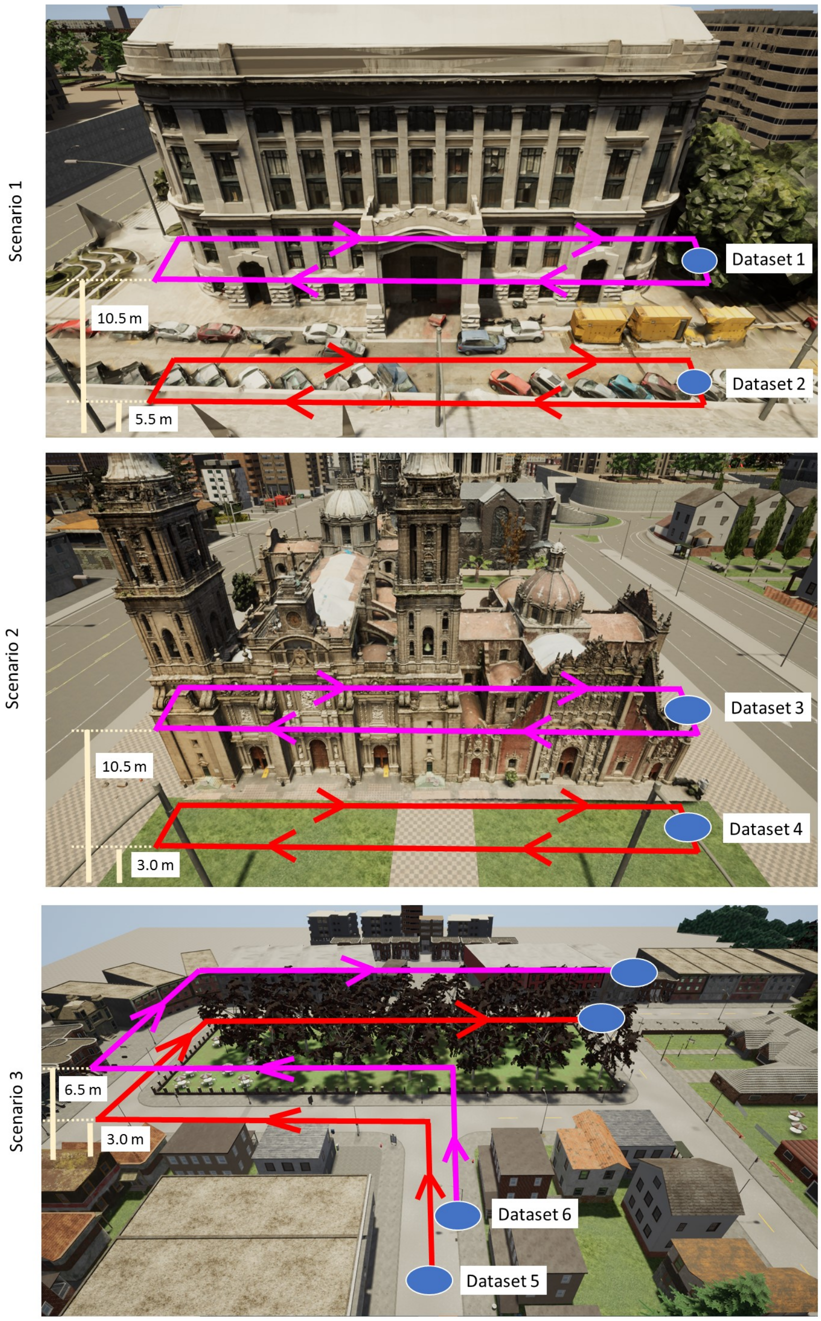


Figure 3.6: Panoramic views of the virtual scenarios and trajectories followed by the cameras used to create the datasets from CARLA simulator [23]. (*1st row*) Scenario 1 used for generating synthetic Datasets 1 and 2. (*2nd row*) Scenario 2 used for generating synthetic Datasets 3 and 4. (*3rd row*) Scenario 3 used for generating synthetic Datasets 5 and 6.

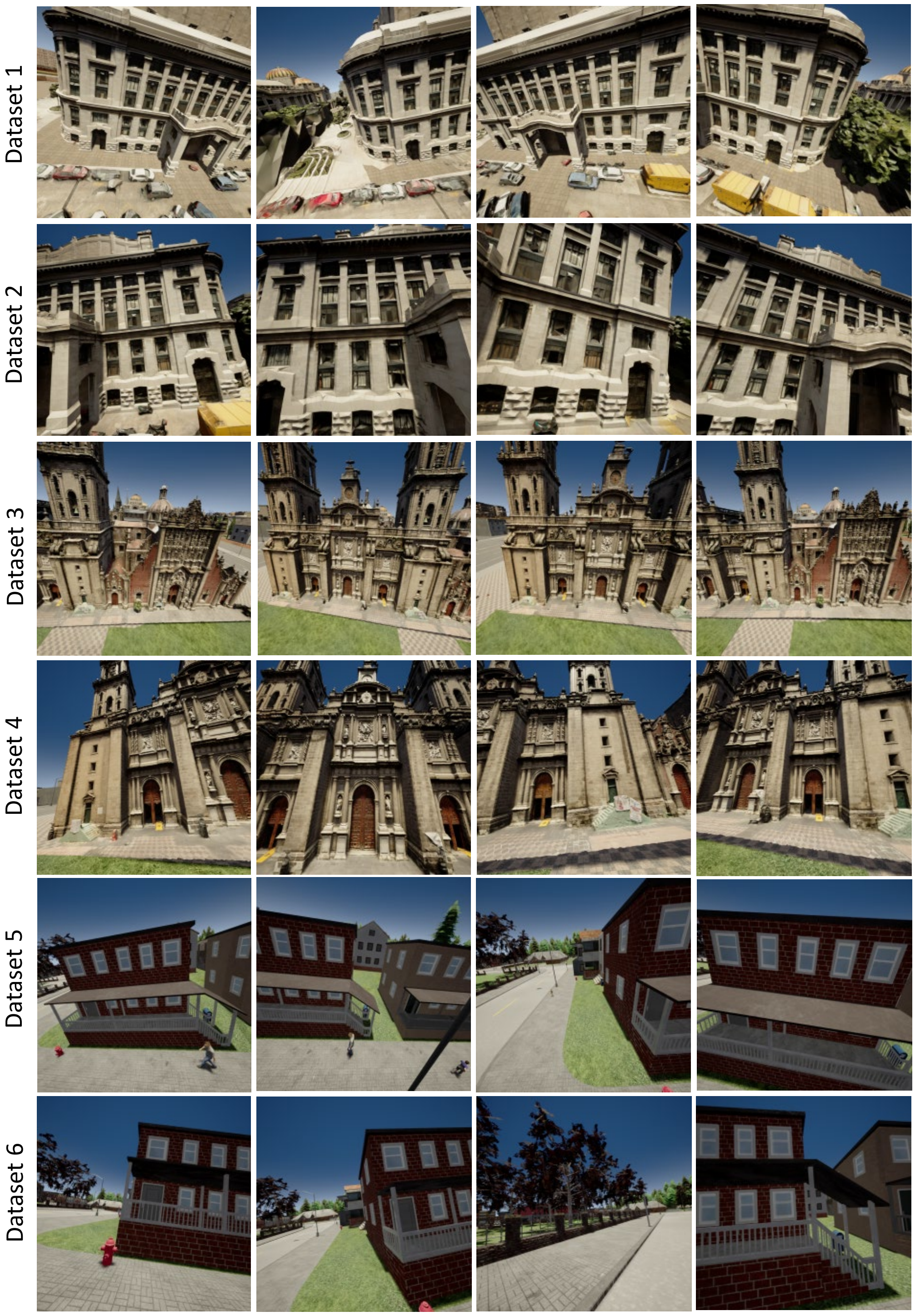


Figure 3.7: Illustration of: Datasets 1 and 2 generated from Scenario 1; Datasets 3 y 4 generated from Scenario 2; Datasets 5 y 6 generated from Scenario 3.

different virtual scenarios. It is necessary to consider that virtual scenario must recreate the real-world scenario using 3D models of existing buildings, peoples or objects, which should be similar to the objects and structures of buildings of real-world scenario. Additionally, the orientation between the cameras and objects of scenario should be also considered.

Figure 3.6 shows illustrations of the buildings used to generate the virtual scenes. The first two scenarios take as a reference the King's College and Old Hospital real-world scenes, while the last scenario is used to demonstrate the importance on the similarity of the geometry of the objects contained in the scene; the trajectories followed by the cameras are also depicted. These trajectories always start from the blue point and move through the red or fuchsia lines. The aforementioned sequence is executed a number of times depending on the virtual scene. At the same time that the configured cameras follow the predefined trajectory, their pose randomly varies at each instant of time (i.e., on the translation the X coordinate could take random values in between [-3, 3]m, the Y coordinate [-1, 3.5]m, while the Z coordinate [-1, 1.5]m; regarding the orientations, the pitch angle could take random values in between [-3.5, 3.5] degrees, yaw angle [-15, 15] degrees while roll angle [-15, 15] degrees). These values were defined according to the characteristics of each scene keeping values that guarantee a large overlap between the views.

In order to evaluate the importance on similarity between synthetic and real image datasets, six synthetic image datasets have been generated from the virtual scenarios mentioned above. Figure 3.7 shows some illustrations of these datasets, which are intended to evaluate the importance of geometry similarity between the virtual and real environments as well as the camera pose similarity.

### 3.3.2 Domain Adaptation Strategy

The cost of labeling data is an expensive and time consuming task needed for most learning based approaches. An alternative solution is to consider datasets and/or models available in similar domains to the problem to be solved (referred to as transfer learning). Domain Adaptation (DA) is a particular case of transfer learning (TL) that leverages labeled data in one or more similar source domains that allow learn important features to apply them in a target domain. In general, it is assumed that the source domains and target domain are related, but not identical, i.e., the train and test data are drawn from the same distribution and share similar joint probability distributions. When the distribution of both source and target domains do not match then the performance of models can be significantly degraded. In real life applications, this constraint might not be kept since the dataset for training and testing could be originated from different features spaces or distributions. This distribution difference is known as domain shift, and is common in real applications. Some changing conditions could increment this domain shift such as background, location, pose changes, even it might be more severe when the geometry of the objects used for both domains are different as well as their appearance—i.e., color and texture (see. Fig. 3.8); for instance, types of images captured from other features spaces or distributions such as photos, painting or sketches [8, 19, 57, 93].

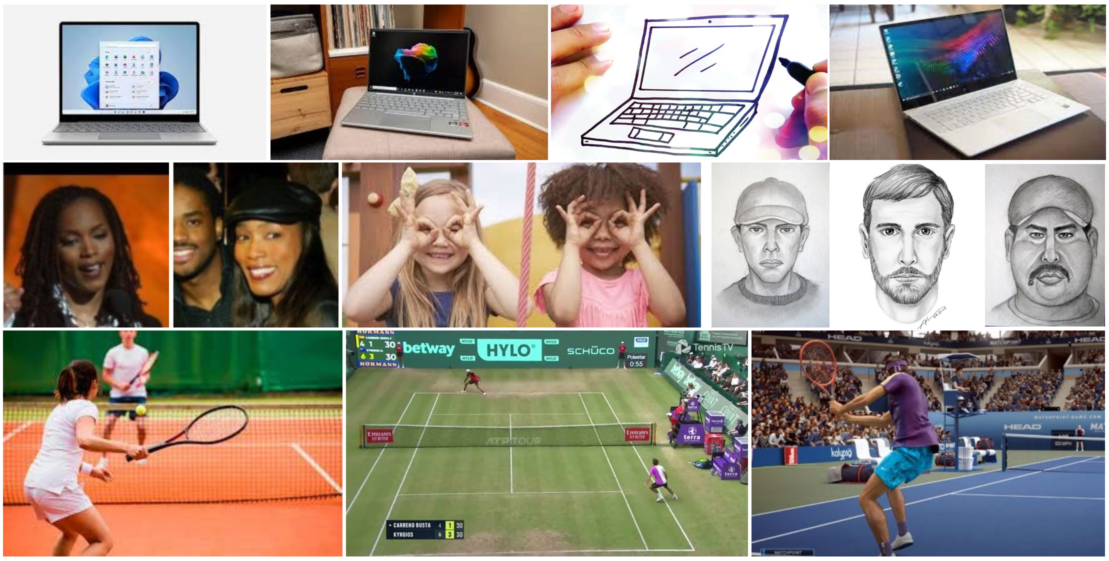


Figure 3.8: (*1st row*) Images of the same object from different sources for object recognition task. (*2nd row*) Face images for face recognition task. (*3rd row*) Tennis sport images for action recognition task.

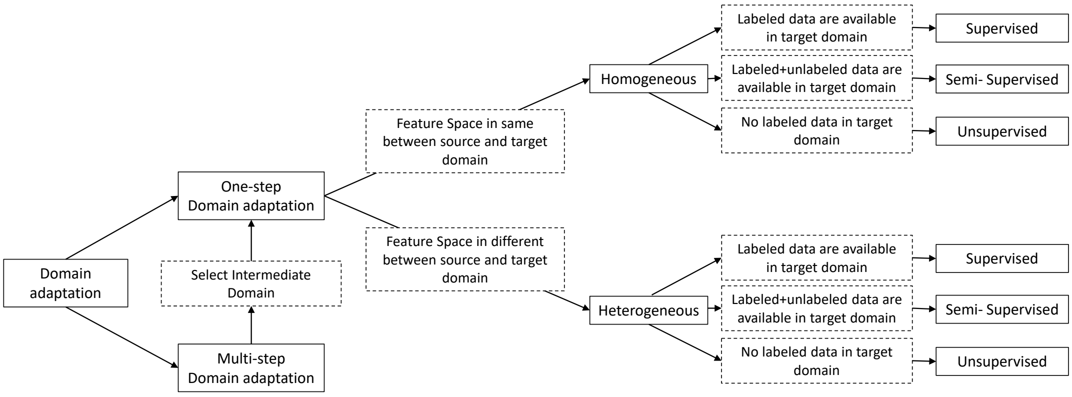

Domain adaptation
One-step Domain adaptation
Select Intermediate Domain
Multi-step Domain adaptation
Feature Space in same between source and target domain
Feature Space in different between source and target domain
Homogeneous
Heterogeneous
Labeled data are available in target domain
Labeled+unlabeled data are available in target domain
No labeled data in target domain
Supervised
Semi- Supervised
Unsupervised
Labeled data are available in target domain
Labeled+unlabeled data are available in target domain
No labeled data in target domain
Supervised
Semi- Supervised
Unsupervised

Figure 3.9: Different settings of Domain Adaptation [120].

The difficulty of collecting datasets according to the target domain when the datasets available do not correspond to the same domain, could be expensive and time consuming due to the amount of human effort involved in. To tackle this problem, the domain adaptation is considered. Its objective is minimized the difference between the source and target domains, making that these learned features of datasets can be generalized to the new domain.

Figure 3.9 shows two main categories based on different domain divergences (distribution shift or feature space difference): homogeneous and heterogeneous. Considering labeled data of the target domain, DA can be categorized into supervised, semi-supervised and unsupervised learning.

Considering the mentioned above, and the synthetic image datasets generated from different virtual environments, DA strategy for supervised learning is considered in the current work. Since both datasets (i.e., the synthetic images and the real images datasets) could have different feature spaces and distribution, transferring the knowledge learned from virtual to real environments can be performed in one-step or multi-step DA. For the first case, both domains are directly related since the features spaces are similar. In the second case, an intermediate domain, which should be highly related with both domains, is necessary.

Figure 3.10 shows an illustration of this DA strategy proposed in this thesis. Three different architectures are used to evaluate this strategy; first, RelPoseTL architecture proposed in Section 3.2. All layers are initialized up to the fourth residual block with the weights of Resnet-50 pretrained on ImageNet dataset. For the remaining layers, the normal distribution is used. Adam optimizer [136] is used to train the network with a learning rate of $10^{-4}$ and batch size of 32. The $\widehat{s}_x$ and $\widehat{s}_y$ variables, eq. (3.9), are initialized with -3.0 and -6.5 in all the experiments respectively; second, an architecture referred to as RPNet$^+$ [26], which is proposed to tackle the relative camera pose problem. The layers are initialized with the weights of the GoogLeNet architecture [105] pretrained on Place365 dataset. The stochastic gradient descent with momentum (SGDM) [104] is used as optimizer with a base learning rate of $10^{-5}$, decay rate of 0.90 each 80 epochs and batch size of 32; and finally, regarding PoseNet architecture [54], it is a slightly modified version of the GoogLeNet architecture. Basically, the softmax layers are removed to output a pose vector of 7-dimensions (position and orientation). SGDM optimizer is used. The scale factor $\beta$, is used in the loss function to keep the position and orientation errors at the same range.

The DA strategy mentioned above is used to train these network architectures with six different synthetic image datasets, which were generated from different virtual environments (see Fig. 3.6). These synthetic images were captured by different cameras from different orientations at the same time. Fig. 3.11 and Fig. 3.12 show real-images datasets with its respective synthetic images, while that Fig. 3.13 only shows synthetic images of a totally different environment with respect to real-image datasets, which are used to apply the DA strategy. As a pre-processing stage, images were resized to 224×224 pixels, including data normalization, like in the real image case. For the training process, 8192 pairs of images were considered to train the RelPoseTL and RPNet$^+$ architectures; while the PoseNet architecture has been trained with just 900 images. It should be mentioned that PoseNet needs just a set of single images, instead of a set of pair of images, like the previous architectures. Each of the synthetic image datasets contain 900 images. The training process was performed for all synthetic image datasets until convergence, which approximately took 30 hours each one for RelPoseTL and RPNet$^+$, and approximately 20 hours each one for PoseNet. The pre-processing mentioned above was also used during the evaluation phase. In the evaluation, a set of 2048 pairs of images from each synthetic image dataset has been considered for RelPoseTL and

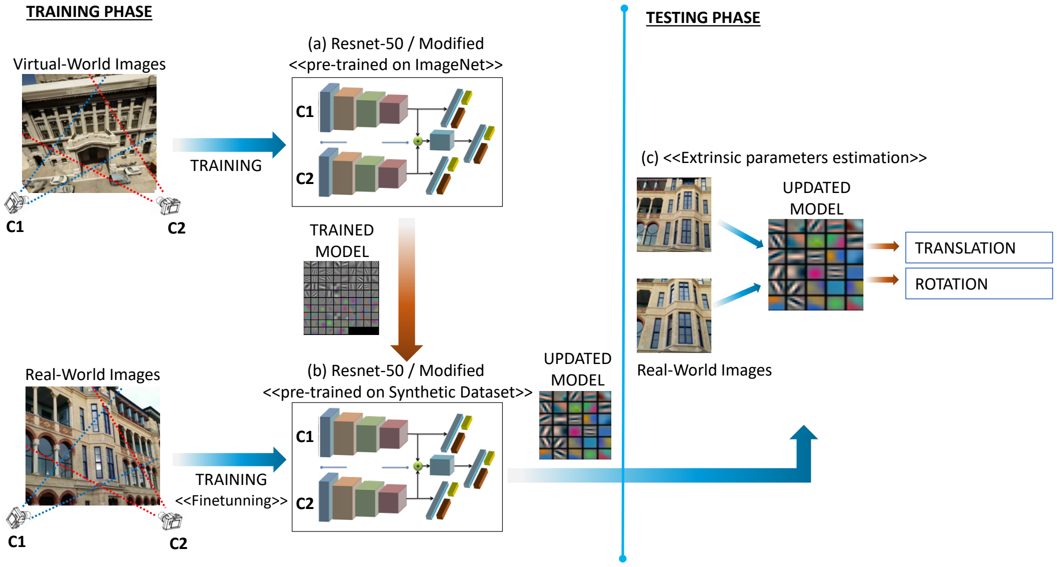


Figure 3.10: (a) RelPoseTL is trained using synthetic images generated from CARLA Simulator. (b) The learned knowledge is used to apply DA strategy using real images. (c) Updated weights after DA strategy are used to estimate relative camera pose (i.e., relative translation and rotation).

RPNet$^+$ architectures, while just 150 synthetic images for the PoseNet architecture.

Finally, the DA strategy is applied to transfer the knowledge learned in the synthetic image domain to the new domain (i.e., real-world). This strategy is also referred in the literature to as transfer learning. It consists on training again the RelPoseTL architecture but now with just a few pairs of real-world images; the learned weights from the synthetic image domain are used as initialization of all layers. The transfer of knowledge is applied by retraining and refining each network, in an end-to-end way, with real-world images; this process is referred to as fine-tunning. The fine-tunning process is independently performed for each one of the synthetic image datasets used to train the architectures mentioned above. In each fine-tunning process three sets of 256, 512, and 1024 pairs of real-world images are considered, which belong to the Old Hospital and King's College of Cambridge Landmark dataset [54]. These three sets of pairs of real-world images are used to demonstrate that the DA strategy could help to obtain appealing results, even using a few pairs of images for training process, and that they could improve if DA strategy is also considered with more set of pairs of real-world images. These processes take about 60, 90, and 120 minutes for each set of pairs of real-world images for RelPoseTL architecture, and 90, 180 and 360 minutes respectively for RPNet$^+$ and PoseNet architectures, until convergence is reached.

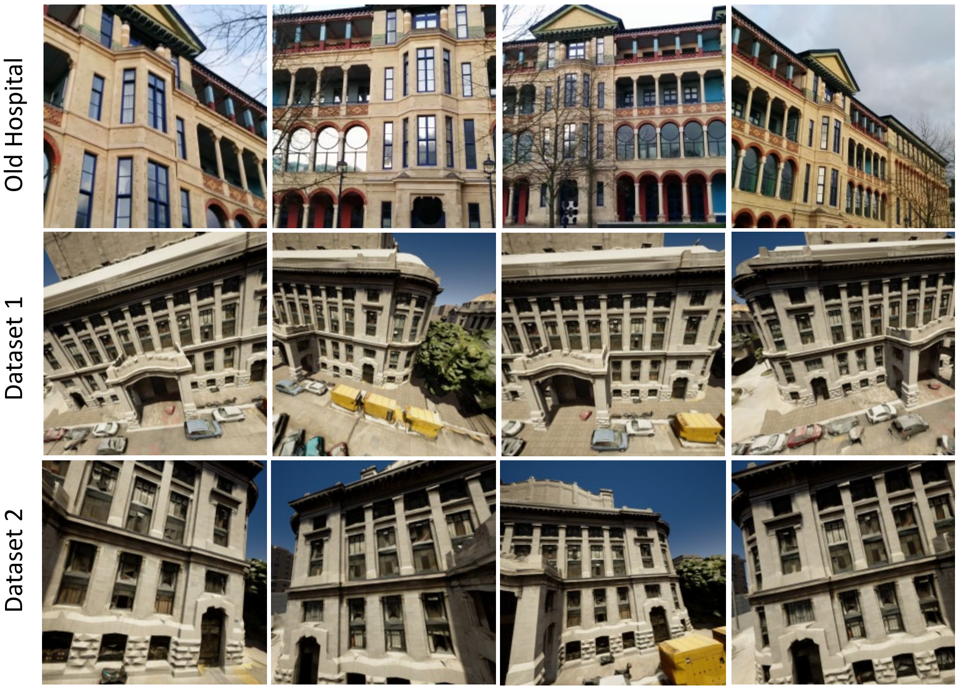


Figure 3.11: *(1st row)* Real-world images, Old Hospital of Cambridge dataset [54]. *(2nd and 3rd row)* Synthetic image datasets generated by the CARLA simulator [23] from different points of views.

### 3.3.3 Experimental Results

The quantitative evaluations and comparison of applying DA strategy are presented in this section. The RelPoseTL, RPNet$^+$ and PoseNet architectures mentioned above are trained by using just pairs of real images, and compared with the same architectures but trained using DA strategy, which consider synthetic images datasets obtained from the CARLA simulator. This comparative is performed to determinate if, irrespective of the used architecture, the results can be improved using the DA strategy. In details, all layers of these architectures are initialized as presented in Section 3.3.2, and trained using just real image dataset. As a pre-processing stage for all these architectures, the real images are resized to 224 pixels along the shorter side; then, the mean and standard deviation are applied as a normalization process to each image. In the training process, three sets of 256, 512, and 1024 pairs of real-world images are used, which are obtained from King's College and Old Hospital of Cambridge dataset. For each set, random crops of 224×224 pixels are computed to feed the network architectures, since it allows to better generalize the training. All sets of data are used to train the networks until convergence, which approximately took 60, 90, and 120 minutes respectively for RelPoseTL, and 90, 180 and 360 minutes respectively for RPNet$^+$ and PoseNet.

Table 3.4: Euclidean distance and angular errors for the DA strategy on pairs of images (PoI) from Old Hospital of Cambridge dataset. In each case the network has been initially trained with the six synthetic datasets (DTi: Dataset i; RD: Real Data). The best results for each network are highlighted with boldface and the corresponding training strategy with lightgray color.

| | | DA strategy on Old Hospital dataset | | |
|---|---|---|---|---|
| | Trained<br>with | Train: 256 PoI<br>Test: 64 PoI | Train: 512 PoI<br>Test: 128 PoI | Train: 1024 PoI<br>Test: 256 PoI |
| RelPoseTL [10] | RD | 4.29m, 5.72$^o$ | 3.93m, 4.04$^o$ | 3.48m, 3.95$^o$ |
| RelPoseTL [10] | DT1 + RD | 4.16m, 6.43$^o$ | 3.61m, 4.23$^o$ | 3.66m, 3.99$^o$ |
| | DT2 + RD | **3.55m, 5.59$^o$** | **3.40m, 3.70$^o$** | **3.20m, 3.54$^o$** |
| | DT3 + RD | 3.61m, 6.53$^o$ | 3.59m, 4.25$^o$ | 3.31m, 3.72$^o$ |
| | DT4 + RD | 3.69m, 5.59$^o$ | 3.45m, 4.33$^o$ | 3.29m, 4.31$^o$ |
| | DT5 + RD | 4.82m, 6.35$^o$ | 4.12m, 5.14$^o$ | 3.94m, 4.65$^o$ |
| | DT6 + RD | 4.57m, 7.87$^o$ | 3.98m, 6.12$^o$ | 4.11m, 5.89$^o$ |
| RPNet$^+$ [26] | RD | 4.06m, 5.70$^o$ | 3.07m, 5.63$^o$ | 3.04m, 5.08$^o$ |
| RPNet$^+$ [26] | DT1 + RD | 3.19m, 5.74$^o$ | 3.06m, 5.08$^o$ | 3.05m, 5.30$^o$ |
| | DT2 + RD | **3.69m, 5.13$^o$** | **3.07m, 5.04$^o$** | **3.03m, 4.57$^o$** |
| | DT3 + RD | 3.21m, 5.55$^o$ | 3.11m, 5.13$^o$ | 3.04m, 5.04$^o$ |
| | DT4 + RD | 4.01m, 6.08$^o$ | 3.08m, 5.17$^o$ | 3.22m, 4.78$^o$ |
| | DT5 + RD | 3.23m, 5.78$^o$ | 3.08m, 5.06$^o$ | 3.13m, 4.67$^o$ |
| | DT6 + RD | 3.57m, 6.05$^o$ | 3.11m, 5.44$^o$ | 2.95m, 4.86$^o$ |
| PoseNet [54] | RD | 3.28m, 3.56$^o$ | 3.75m, 5.01$^o$ | 3.69m, 5.38$^o$ |
| PoseNet [54] | DT1 + RD | 2.95m, 3.82$^o$ | 3.31m, 4.83$^o$ | 3.68m, 5.29$^o$ |
| | DT2 + RD | **3.19m, 3.39$^o$** | **3.09m, 4.73$^o$** | **3.55m, 5.33$^o$** |
| | DT3 + RD | 2.49m, 3.64$^o$ | 3.55m, 4.72$^o$ | 3.71m, 5.23$^o$ |
| | DT4 + RD | 3.20m, 3.71$^o$ | 3.83m, 4.60$^o$ | 3.81m, 4.62$^o$ |
| | DT5 + RD | 3.05m, 3.72$^o$ | 3.69m, 4.61$^o$ | 3.58m, 5.33$^o$ |
| | DT6 + RD | 3.26m, 3.44$^o$ | 3.72m, 4.67$^o$ | 3.87m, 5.43$^o$ |

Table 3.5: Euclidean distance and angular errors for the DA strategy on pairs of images (PoI) from King's College of Cambridge dataset. In each case the network has been initially trained with the six synthetic datasets (DTi: Dataset i; RD: Real Data). The best results for each network are highlighted with boldface and the corresponding training strategy with lightgray color.

| | | DA strategy on King's College dataset | | |
|---|---|---|---|---|
| | Trained with | Train: 256 PoI<br>Test: 64 PoI | Train: 512 PoI<br>Test: 128 PoI | Train: 1024 PoI<br>Test: 256 PoI |
| RelPoseTL [10] | RD | 5.28m, 5.29° | 3.86m, 5.08° | 2.95m, 4.06° |
| RelPoseTL [10] | DT1 + RD | 4.95m, 8.31° | 4.14m, 4.32° | 3.38m, 3.75° |
| | DT2 + RD | 4.92m, 5.50° | 3.72m, 4.18° | 2.86m, 3.68° |
| | DT3 + RD | 4.94m, 5.63° | 3.99m, 4.93° | 3.04m, 3.82° |
| | DT4 + RD | **4.89m, 4.96°** | **3.13m, 4.18°** | **2.35m, 3.32°** |
| | DT5 + RD | 5.07m, 8.30° | 4.07m, 6.14° | 3.85m, 4.94° |
| | DT6 + RD | 4.81m, 7.94° | 4.89m, 7.16° | 3.94m, 5.99° |
| RPNet$^+$ [26] | RD | 1.90m, 4.35° | 1.58m, 3.87° | 1.48m, 3.38° |
| RPNet$^+$ [26] | DT1 + RD | 2.39m, 4.16° | 1.95m, 3.69° | 1.49m, 2.98° |
| | DT2 + RD | 2.30m, 3.93° | 1.62m, 3.11° | 1.48m, 2.73° |
| | DT3 + RD | 2.33m, 4.19° | 1.84m, 3.52° | 1.52m, 3.03° |
| | DT4 + RD | **1.85m, 3.44°** | **1.54m, 3.00°** | **1.27m, 2.69°** |
| | DT5 + RD | 2.13m, 4.09° | 1.82m, 3.38° | 1.62m, 3.30° |
| | DT6 + RD | 2.35m, 3.74° | 1.77m, 3.23° | 1.51m, 2.83° |
| PoseNet [54] | RD | 2.14m, 3.92° | 2.13m, 3.14° | 2.24m, 2.76° |
| PoseNet [54] | DT1 + RD | 2.31m, 4.64° | 2.29m, 3.65° | 2.36m, 3.03° |
| | DT2 + RD | 1.83m, 3.73° | 2.03m, 3.16° | 1.99m, 3.17° |
| | DT3 + RD | 2.38m, 3.84° | 2.40m, 3.56° | 2.24m, 3.34° |
| | DT4 + RD | **1.95m, 3.55°** | **1.91m, 2.99°** | **1.83m, 2.74°** |
| | DT5 + RD | 2.27m, 4.63° | 2.12m, 3.15° | 2.32m, 2.85° |
| | DT6 + RD | 2.26m, 3.98° | 1.96m, 3.07° | 1.95m, 2.98° |

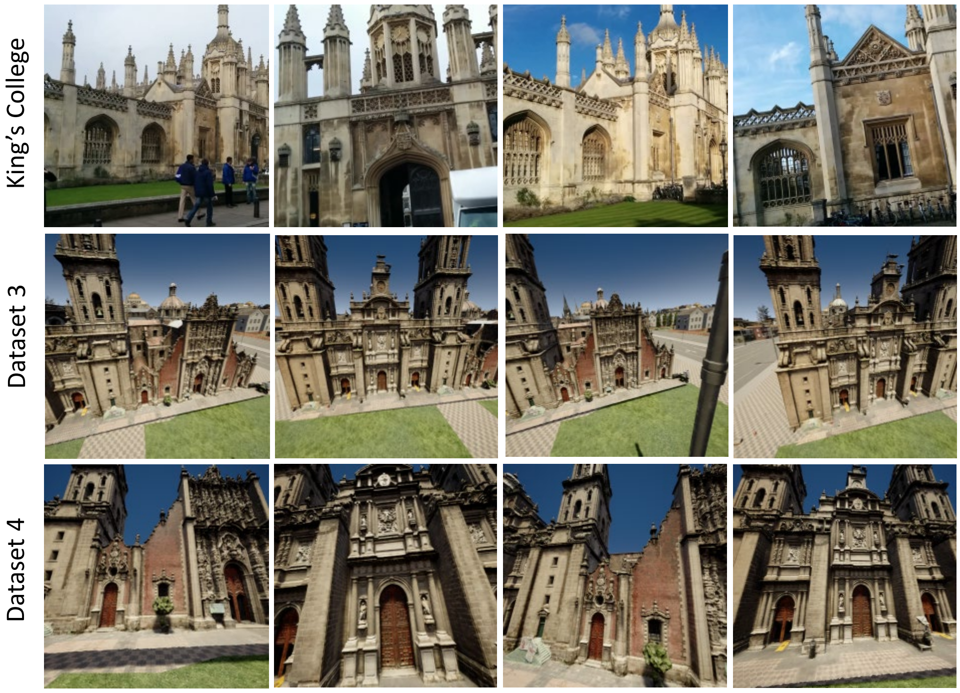


Figure 3.12: *(1st row)* Real-world images, King's College of Cambridge dataset [54]. *(2nd* and *3rd row)* Synthetic image datasets generated by the CARLA simulator [23] from different points of views.

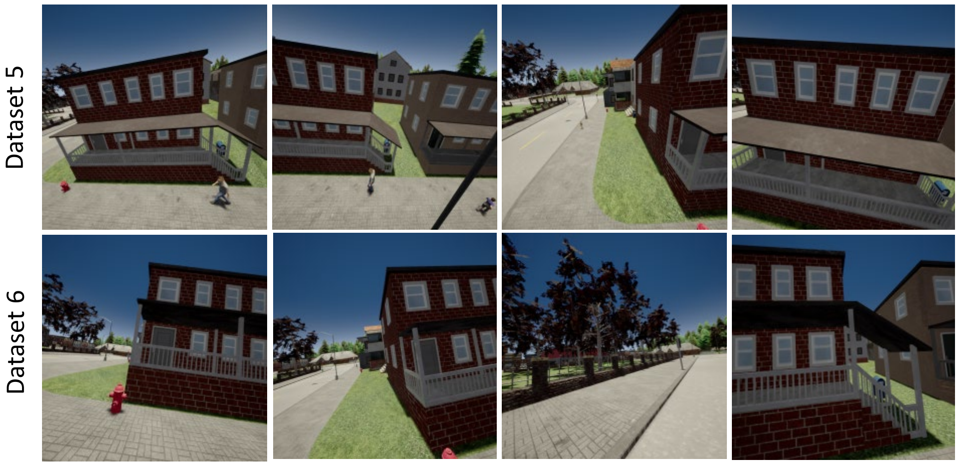


Figure 3.13: Synthetic image datasets generated by the CARLA simulator [23]. These datasets have no similarity to Old Hospital or King's College datasets.

For a fair comparison between both cases (with/without the usage of the proposed DA strategy), in the evaluation phase, three sets of 64, 128, and 256 pairs of real-world images are used, which are obtained from King's College and Old Hospital of Cambridge dataset. The pre-processing mentioned above is also used during the evaluation phase. However, on the contrary to the training phase, central crops is used instead of random crop, since this strategy is generally used in the literature [26, 52, 54]. Table 3.4 and Table 3.5 present the errors obtained for each case. Angular error and Euclidean distance error are used to evaluate the performance of the architectures with/without the usage of the proposed DA strategy. All the datasets presented in Section 3.3.1 are considered to train, together sets of real-world images mentioned in Section 3.3.2, different experiments using architectures such as RelPoseTL, RPNet$^+$ and PoseNet. The evaluation of DA strategy for these experiments is performed using the three sets of pairs of real-world images mentioned above. Angular error is used to compute the errors between the estimated rotation and the ground truth, which is represented as a quaternion (i.e., a 4-dimensional vector). On the other hand, the Euclidean distance is used to measure the error of translation between the estimated by the architecture, which is represented as a 3-dimensional vector, and the ground truth. The proposed DA strategy is applied over each of the synthetic image datasets to evaluate the importance of scene geometry similarity as well as camera pose used for the dataset acquisition between the images captured from a real scenario and virtual scenario.

As it can be appreciated in Table 3.4, the best results in the case of Old Hospital dataset are obtained with the DA strategy, i.e., when knowledge learned using synthetic images from Dataset 2 are considered to initialize and start the training of all architectures mentioned above. With Dataset 2, and by using the proposed DA strategy, better results are obtained in all the cases (256, 512, or 1024 pairs of real images), both translation and rotation, if they are compared with the architectures trained with just pairs of real images. It should be highlighted that Dataset 2 is the most similar to the Old Hospital dataset (see images in Fig. 3.11 (*1st row* and *3rd row*)). Dataset 1 (see images in Fig. 3.11 (*2nd row*)) has been acquired in the same virtual scenario than Dataset 2. However, the cameras were placed further from the building and on a top view. Regarding the King's College dataset, also better results are reached when the proposed DA strategy, using synthetic images, is considered. In this case, the best results correspond to the DA strategy with Dataset 4. Synthetic images in Dataset 4 are the most similar to the real images in King's College dataset (see images in Fig. 3.12 (*1st row* and *3rd row*)). Dataset 3 (see images in Fig. 3.12 (*2nd row*)) has been acquired in the same virtual scenario than Dataset 4, but in this case the cameras were placed further from the building and on a top view.

The quantitative improvements with the proposed DA strategy (i.e., Dataset 2 for the Old Hospital dataset and Dataset 4 for the King's College dataset for all network architectures) are as follow. For Old Hospital dataset, the following improvements are reached; first, RelPoseTL arquitecture [10]: $i$) about 17% on translation and 2% on rotation for a set of 256 pairs of real-world images; $ii$) about 13% on translation and 8% on rotation for a set of 512 pairs of real-world images; and $iii$) about 8% on translation and 10% on rotation for a set of 1024 pairs of real-world images. Second, on the RPNet$^+$ architecture [26] the following improvements have been reached: *i*) about 9% on translation and 10% on rotation for a set of 256 pairs of real-world images; *ii*) on translation there is no improvement and about 10% on rotation for a set of 512 pairs of real-world images; and *iii*) about 0.3% on translation and 10% on rotation for a set of 1024 pairs of real-world images. Finally, on PoseNet architecture [54]: *i*) about 3% on translation and 5% on rotation for a set of 256 pairs of real-world images; *ii*) about 18% on translation and 6% on rotation for a set of 512 pairs of real-world images; and *iii*) about 4% on translation and 0.9% on rotation for a set of 1024 pairs of real-world images.

With respect to the second case, the King's College dataset, the following improvements have been reached; first, with the RelPoseTL architecture [10]: *i*) about 7% on translation and 6% on rotation for a set of 256 pairs of real-world images; *ii*) about 19% on translation and 18% on rotation for a set of 512 pairs of real-world images; and *iii*) about 20% on translation and 18% on rotation for a set of 1024 pairs of real-world images. Second, on the RPNet$^+$ architecture [26]: *i*) about 3% on translation and 21% on rotation for a set of 256 pairs of real-world images; *ii*) about 3% on translation and 22% on rotation for a set of 512 pairs of real-world images; and *iii*) about 14% on translation and 20% on rotation for a set of 1024 pairs of real-world images. Finally, on PoseNet architecture [54] the following improvements have been reached: *i*) about 9% on translation and 9% on rotation for a set of 256 pairs of real-world images; *ii*) about 10% on translation and 5% on rotation for a set of 512 pairs of real-world images; and *iii*) about 18% on translation and 0.7% on rotation for a set of 1024 pairs of real-world images.

The results obtained by the PoseNet architecture after applying the proposed DA strategy, are slightly better compared with the architecture trained with just single images. This little improvement could be explained by the small number of synthetic images used for train the PoseNet architecture in the proposed DA strategy; as mentioned in Section 3.3.2, this architecture needs single images, instead of pair of images that use the other two architectures, hence all the images contained in the synthetic image datasets have been considered. This suggests the importance on the number of synthetic images used by the proposed DA strategy. Additionally, the results shown in Table 3.4 and Table 3.5, indicate that regardless of the architecture, the results are improved when the DA strategy is applied.

The importance of the similarity between both image datasets, i.e., real and synthetic images, which could be analyzed from the results presented in Table 3.4 and Table 3.5 when Dataset 5 and Dataset 6 are considered. These two datasets have been acquired in a synthetic scenario completely different from the Old Hospital and King's College of Cambridge dataset (see Fig. 3.13). None of the networks pre-trained with Dataset 5 or Dataset 6 reach the best result. As mentioned above, in each case the best results have been obtained when the most similar datasets are considered. This similarity should include both the virtual environment (3D scenario used to represent the real environment) as well as the trajectories of the cameras when the synthetic images are acquired, i.e., the distance between the cameras and the objects and the relative camera-object orientation, both features matter.

## 3.4 Conclusions

This chapter focuses on answering one of the research questions, which is the relative camera pose estimation from multi-view system, mainly when there is a lack of large datasets for algorithm training. The proposed architectures for solving the relative camera pose estimation problem are Siamese neural networks, which take as input a set of pairs of images. These pairs of images are captured from a given scenario at the same time from multi-view system. However, due to the scarcity of large dataset of real-world images for training these architectures, a domain adaptation strategy is proposed. The proposed DA strategy leverages the synthetic image datasets captured from virtual environments generated by CARLA Simulator to overcome the dependency on real-image datasets, especially, when these are scarce. Likewise, the proposed approach shows how the features extracted on the synthetic images could help to have a better approximation of the weights of the network, and then, to adapt it to the real-world images to reduce the translation and rotation errors. Experimental results and comparisons are provided showing improvements on the obtained results (relative rotation and translation). However, to take full advantage of the proposed approach, the virtual environments' contents should have similar features with respect to the real environments' contents. Furthermore, not only the environments' contents should be similar to the real scenario, but also the images should be acquired in a similar way (i.e., distance and point of view between camera and objects).

# 4 Human Pose Estimation

This chapter focuses on answers how the relative camera pose from multi-view environments could help to estimate occluded human body parts, as well as if using the relative camera pose, the accuracy of human pose estimation from multi-view environments could be increased, being two of the research questions propose in this thesis. The problems are tackled by proposing two novel architectures that take advantage of the relative camera pose to improve results, especially to tackle the self occlusions cases. Other approach based on attention modules for human pose estimation problem is also developed and evaluated.

## 4.1 Introduction

The 2D Human Pose Estimation problem has been studied during the last decade. Basically, this problem is generally tackled by first detecting human body joints such as head, elbow, knee, shoulder, etc, and then connecting them to build the human body skeleton. The solutions proposed in the state-of-art such as OpenPose [5], DeepPose [112], Stacked Hourglass Networks [76], among others, are robust when all body joints are detected (see Chapter 2 for more details). However, when joints are occluded due to natural human body pose (i.e., self-occlusions) or certain object in the scene (e.g., bicycles, cars), it become a challenging problem, mainly, in monocular vision systems. Some applications such as human action recognition, augmented reality, surveillance, gaming, healthcare, take advantage of the accuracy of 2D human pose to develop on top of them different solutions.

Nowadays, most of computer vision tasks: for instance, image enhancement, object detection, camera pose estimation, just to mention a few, have been tackled using CNNs, reaching a better result with respect to classical approaches [5, 27, 76, 102, 110, 125, 127]. Different proposals have been designed to tackle the human pose estimation problem from monocular vision system scenarios by using CNN architectures. The input of these architectures is a set of images with single o multi-person, which have been captured from one camera. In the case of multi-persons input data, the computational cost could be increased due to the number of body joints of each subject that the architecture has to detect in the image.

Other approaches proposed during the last years are the transformers, which have attention mechanisms as an integral part of compelling sequence modeling and transduction models in various tasks. Although the transformers have been designed to tackle problems of Natural Language Processing, its attention mechanisms have been also applied in some computer vision tasks. They have helped to pay more attention to important areas and suppress other unnecessary information. They have been widely applied in tasks such as object detection [6, 140], segmentation [118, 123], low-level vision task [13], including both 3D hand pose estimation and 3D human pose estimation [41, 42, 59], showing that are suitable for modeling human pose.

Although the results obtained from monocular vision systems for the human pose estimation problem are appealing, the occlusions of the human body joints are challenging problems, which have not been completely solved. The problem could be tackled from a multi-view approach since the human body is captured from different points of view at the same time. This could allow to recover body joints occluded in one view by using information from other cameras, from another point of view, where these body joints are not occluded.

The multi-view approaches have been already used in some tasks such as camera pose, 3D-reconstruction or object detection [11, 91, 108, 131], where the principal problems are the occluded regions. In this work, the multi-view scenario is considered to tackle the human pose estimation problem, and thus improve the accuracy of body joints, including those that are occluded. The contributions of this chapter are as follow:

- Develop a CNN architecture to tackle the human pose estimation problem from a multi-view scheme that considers the relative camera pose—extrinsic parameters—to fuse the features important from other view, and thus, improve the accuracy of occluded body joints.
- Develop a CNN architecture using attention mechanisms to tackle the human pose estimation problem without taking into account the relative camera pose—extrinsic parameters.
- Show the importance of redundancy of information generated from other views to help to get more accuracy when the body joints are occluded.

This chapter results on the following publications

- **Jorge L. Charco**, Angel D Sappa, and Boris X Vintimilla. Human pose estimation through a novel multi-view scheme. In International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, pages 855–862. INSTICC, SciTePress, 2022.
- **Jorge L. Charco**, Angel D. Sappa, Boris X. Vintimilla, and Henry O. Velesaca. Human Body Pose Estimation in Multi-view Environments. ICT Applications for Smart Cities, pages 79-99. Springer, 2022.

## 4.2 Camera pose based architecture

In order to solve the human pose estimation problem from multi-view environments, a novel architecture is presented, which uses the information geometry of the cameras (i.e., extrinsic parameters) to take advantage of image features from other views. However, it is necessary that the images generated from different cameras, which capture all objects of the scene, should have a minimum of overlapping to ensure that the most of the features of an image can be found in other images.

### 4.2.1 Network Architecture

The proposed architecture leverages the multi-view scheme considering at least two views to tackle the self-occlusion problem in the 2D human pose estimation. The proposed architecture is referred to as Mview-Joints [9]. It uses a CNN backbone proposed by [45], which is a variant of Resnet-152 with learnable weights, and whose output is a 2D-vector that corresponds to the position of body joints on image plane (x,y). These body joints are used to be retrained with the proposed multi-view scheme.

The proposed architecture takes as an input a set of images acquired from a multi-view system. It consists of $C$ calibrated and synchronized cameras with known parameters $R_c$ (i.e., intrinsic and extrinsic parameters), which capture the image of a single-person in the scene from different views. The images acquired by the multi-view system are denoted as $Im^c$, and organized in pairs of images, which belong to different close views, namely, reference view $Im^{ref}$ and source view $Im^{src}$. Heatmaps obtained from each image, like those resulting from the usage of the backbone [45], are fused across source view considering the confidence of each joint and the relative camera pose, improving the accuracy of joints from each image (see Fig. 4.1).

In details, given a set of pairs of images, the CNN backbone extracts the heatmaps of each joint for each input image separately, which are denoted as $M_{\Theta}^{ref} = \left\{Im_1^{ref},...,Im_i^{ref}\right\}$ and $M_{\Theta}^{src} = \{Im_1^{src},...,Im_i^{src}\}$, where heatmaps are referred to as $\Theta$, $i$ is the number of joints and, $ref$ and $src$ correspond to the reference and source view respectively. The heatmaps are used to estimate the 2D positions of each joint for each input image. First, the softmax is computed across the spatial axes; and then, the 2D positions of the joints ($p_{(x,y)}$) are computed as the center of mass of the corresponding heatmaps, which is defined as:

$$p_{(x,y)} = \sum_{u=1}^{W} \sum_{v=1}^{H} h_{i_{(u,v)}}.(\zeta_{\Theta}(h_{i_(u,v)})), \tag{4.1}$$

where $\zeta_{\Theta}$ represents the function softmax; $h$ represents the ROI of the heatmaps of $i$-th joint and W and H correspond to the size of the ROI heatmap. For each 2D position of each joint obtained using Eq. (4.1), its position in the world coordinate system $P = (X, Y, Z)$ is obtained by projecting back the image point to the 3D space.

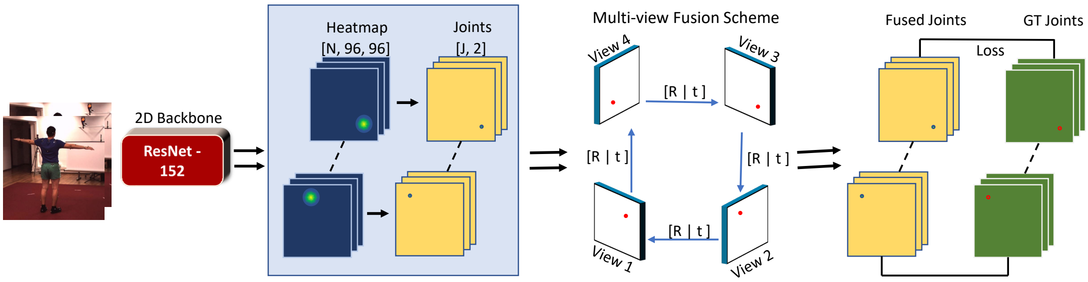


Figure 4.1: CNN backbone is fed with a set of pairs of images of the same scene simultaneously acquired from different points of view. The multi-view fusion scheme allows to estimate occluded joints with information from other views across of the relative camera pose.

$$x_i = f\frac{X}{Z} \qquad y_i = f\frac{Y}{Z}, \tag{4.2}$$

where $(x, y)$ is the 2D position of the $i$-th joint obtained in Eq. (4.1). The focal length of the camera is defined as $f$. Since the depth $(Z)$ of the joint is unknown, a raytracing strategy is followed. It consists on defining two depth values that are used to compute the line going from the camera central point and the studied joint in 3D space using the Eq. (4.2).

The position in the world coordinate system of each joint is computed using the relative camera pose between both points of view (see Fig. 4.2), i.e, source and reference view, and then, projected to the image plane, as shown below:

$$T_{rel} = Rot_{src} \cdot (T_{ref} - T_{src}), \tag{4.3}$$

$$Rot_{rel} = Q(Rot_{ref}.T)^{-1} * Q(Rot_{src}.T), \tag{4.4}$$

$$p^{ref}_{src_{(x,y)}} = \Delta 2D_{ref}(Rot_{rel} \cdot (P_i - T_{rel})), \tag{4.5}$$

where $Q(.)$ represents the quaternion. The rotation matrix and the translation vector are defined as $Rot \in \mathbb{R}^{3\times3}$ and $T \in \mathbb{R}^{3\times1}$ respectively. $P_i$ corresponds at $i$-th joint in the world coordinate system obtained in Eq. (4.2). $\Delta 2D_{ref}(.)$ represents the back-projection of 3D position of $i - th$ joint from camera coordinate system to image plane in the reference view using the intrinsic parameters. The projected line on image plane of reference view $L$ is obtained using the linear equation and the point computed in the Eq. (4.5). In order to obtain the depth of $i$-th joint of the source view, which should be on the projected line on image plane of reference view, the intersection between the projected line and the 2D-point of the joint computed in the reference view $p_{ref_{(x,y)}}$ is obtained.

The confidence of the two different 2D positions in the plane of the reference image of the

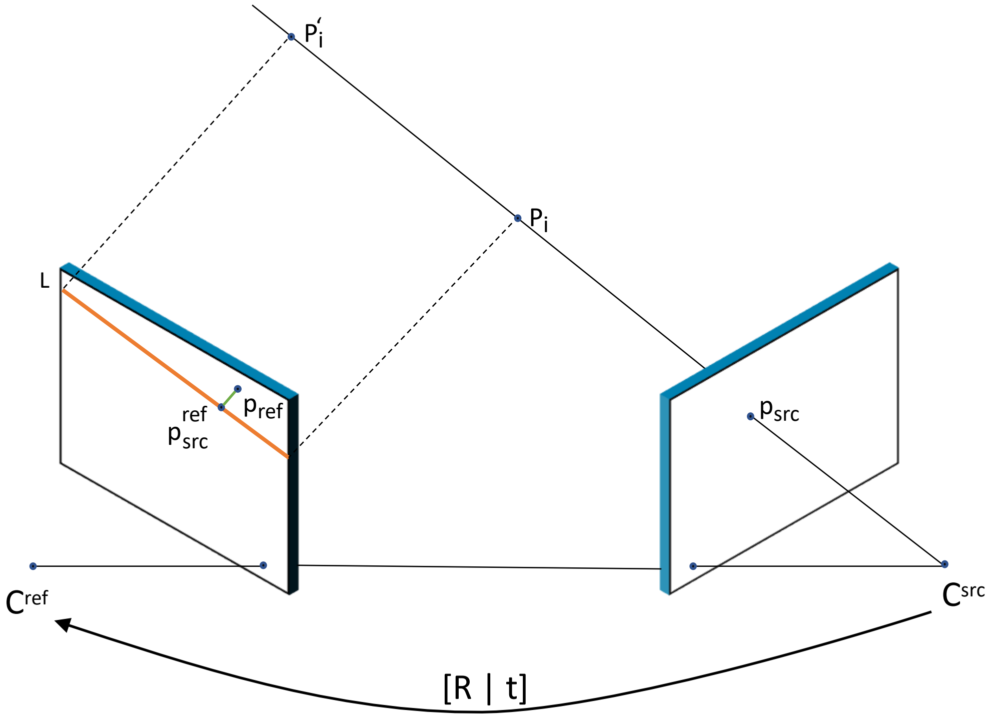


Figure 4.2: An image point $p_{src}$ back-projects to a ray in 3D defined by the point $p_{src}$ and two depth values $p_i$ and $p_i'$; it is then projected to the image plane of reference view to generate the epipolar line $(L)$.

$i$-th joint, where the first corresponds to the reference view $p_{ref_{(x,y)}}$ and the second, a projected joint from source to reference view $p^{ref}_{src_{(x,y)}}$, is computed as the distance between the ground truth of 2D position of $i$-th joint and the estimated 2D positions of $i$-th joints (i.e., $i$-th joint from source and reference views). These confidence values are used as shown in Eq. (4.7).

$$\omega = 1 - \left| \frac{D_\Delta(\hat{\gamma}_i, \gamma_i)}{\sum D_\Delta(\hat{\gamma}_i, \gamma_i)} \right|, \tag{4.6}$$

$$\delta_{upd_{i_{(x,y)}}} = \omega * p_{i_{(x,y)}}, \tag{4.7}$$

where $(\hat{\gamma}\ , \gamma)$ represent the ground truth and prediction of 2D position of $i$-th joint respectively, and $\omega$ corresponds to the confidence of the points of $i$-th joints in the reference view.

The enhanced 2D position of $i$-th joint is denoted as $\delta_{upd_{i_{(x,y)}}}$, which considers the information and confidence of $i$-th joint projected from the source to reference view. In order to minimize the error between the enhanced 2D position of $i$-th joint and its ground truth in the learning process of the proposed multi-view scheme, a loss function is defined as:

$$Loss = \sum_{i=1}^{N} \left\| \delta_{upd_{i_{(x,y)}}} - \hat{p}_{i_{(x,y)}} \right\|_2, \tag{4.8}$$

where $N$ corresponds to the number of joints, and $\hat{p}_{i_{(x,y)}}$ is the ground-truth of $i$-th joint in image plane.

### 4.2.2 Experimental Results

One large-scale pose estimation public dataset is used to conduct the experiments, including the JDR(%) metric and Euclidean distance to evaluate the accuracy from the proposed model. This section will briefly describe both of them, dataset and used metrics, including the obtained results.

#### Dataset

The $Human3.6m$ dataset is proposed by [44], and it is one of the largest publicly available human pose estimation dataset. It can be used with monocular or multi-view setups. The dataset is generated from four synchronized and calibrated digital cameras, which capture 3.6 million frames with a single-person, and they are located in each corner of the room (see Fig 4.3). To ensure the synchronization between the video and pose data, the cameras are on the same clock and trigger as the motion capture system. The motions are performed by 11 professional actors (6 males, 5 female) in different activities such as taking photo, discussion, smoking among other. According to the experiments performed in proposals of the state-of-the-art, subjects 1, 5, 7, and 8 are used for training the proposed approach; while subjects 9 and 11 are used just for testing. Images from all the cameras are used during the training and testing process.

#### Metrics

In the human pose estimation problem, the Joint Detection Rate (JDR) metric is generally used to evaluate the performance of trained model. The JDR metric measures the percentage of $successfully$ detected joints, assuming as a successful detection those joints where the distance between the estimation and the ground truth joint is smaller than a given threshold; in the evaluation of the proposed model, this threshold has been defined as half of the head size, as proposed in [2]. In addition to the JDR metric, the Euclidean distance is computed to measure the error from every estimation with respect to the corresponding ground truth. This allows to know the accuracy of each joint of the estimated human pose. Note that values obtained by JDR metric, somehow hide the accuracy of the estimations, principally, for those joints where the joint estimation is considered as $successful$.

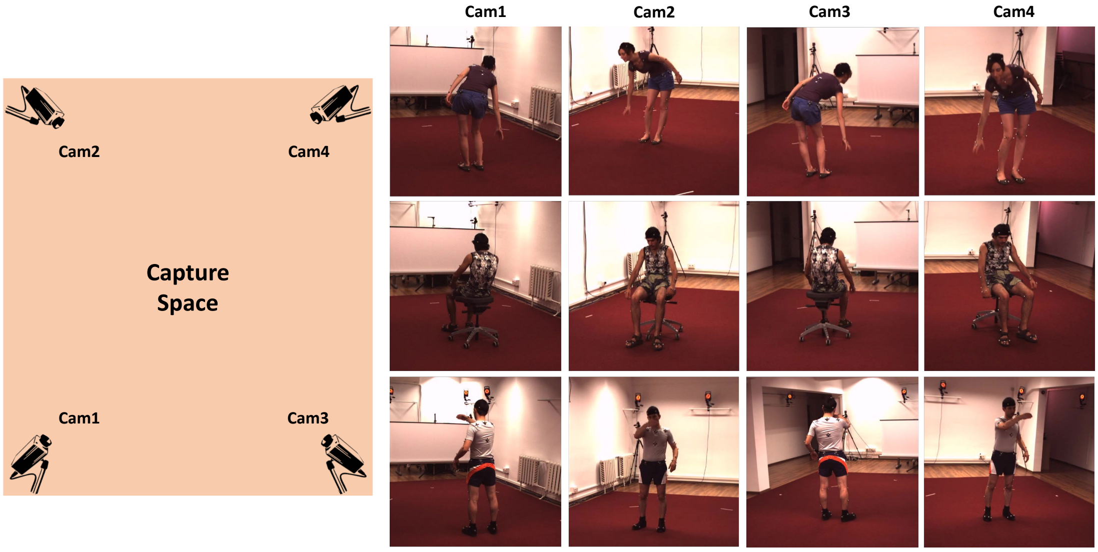


Figure 4.3: Human3.6m dataset [44] used for the training and evaluation processes. The subject is captured from different point of views considering the four calibrated and synchronized cameras located in each corner of the room.

### Training of Multi-View Scheme

The weights of Resnet-152 pretrained by [45] are used to initialized the CNN backbone in the proposed architecture. The network architecture is implemented with Pytorch and trained with NVIDIA Titan XP GPU and Intel Core I9 3.3GHz CPU. Adam optimizer is used to train the network with a learning rate of $10^{-5}$ and batch size of 32 (i.e., eight human poses simultaneously captured from four different points of view). The network architecture is trained on Human 3.6m dataset. As a pre-processing step, the images in the dataset, whose resolutions are 1000×1000, are cropped according to the bounding box, which has all the four sides of equal length with the person in the center for keeping the aspect ratio, and then, resizes them to 384×384 pixels. The mean value of intensity of pixels is computed and subtracted from the images.

For the training process, a set of 60k images are used to feed the network, which is then trained until 20 epochs; it takes about 120 hours. The pre-processing mentioned above has been also used during the evaluation phase. In the evaluation a set of 8k images has been considered.

### Results and Comparisons

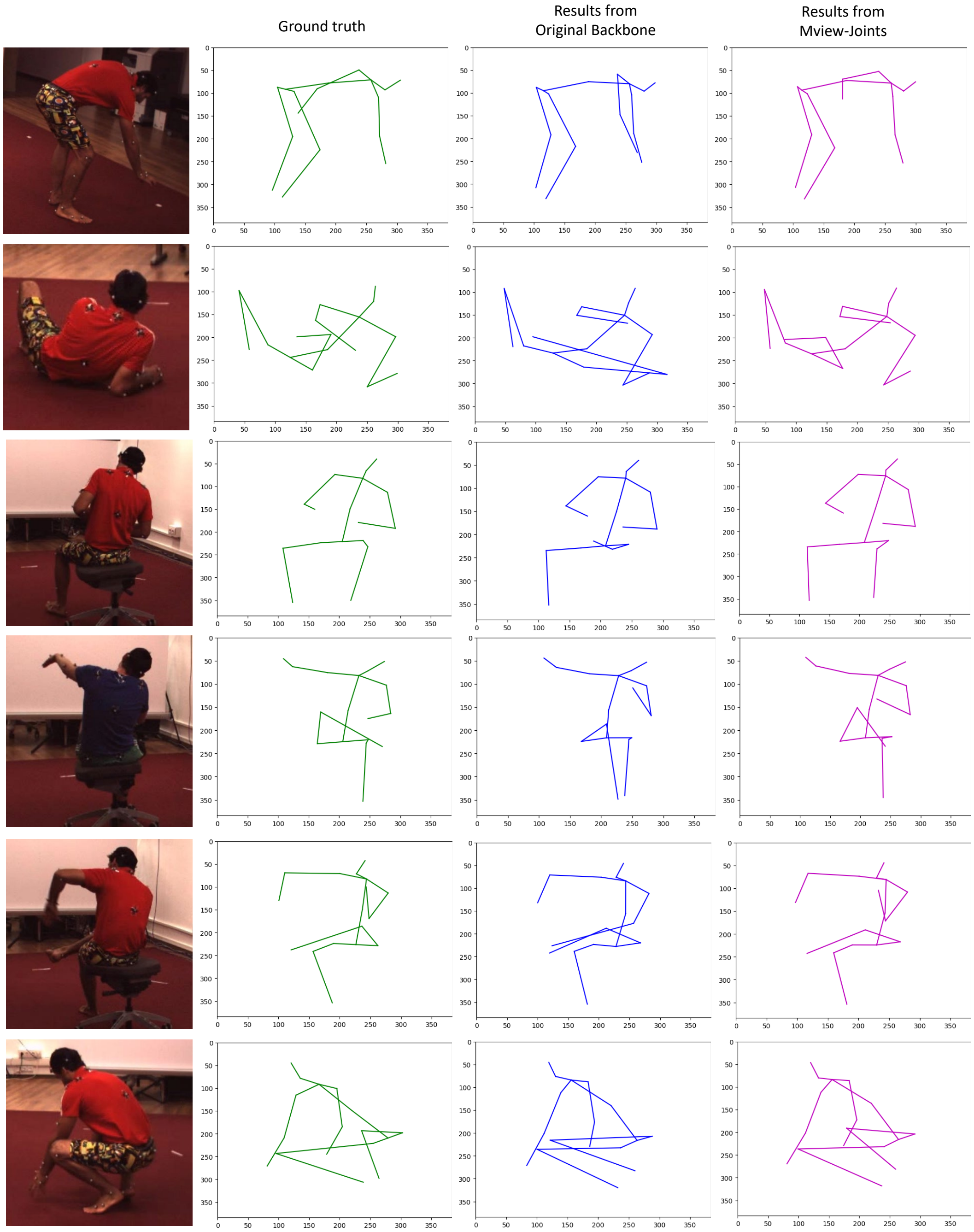
Quantitative results for human pose estimation are depicted in Table 4.1. The evaluations of proposed architecture referred to as Mview-Joints (introduced in Sec 4.2.1) are compared

Table 4.1: Comparison of 2D pose estimation accuracy on Human3.6m dataset using JDR(%) as metric. " – ": these entries were absent. ∗: trained again by [38]. R50 and R152 are ResNet-50 and ResNet-152 respectively. Scale is the input resolution of the network.

| | net | scale | shlder | elb | wri | hip | knee | ankle | root | neck | head | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sum epipolar line [81] | R152 | 320 | 91.36 | 91.23 | 89.63 | 96.19 | 94.14 | 90.38 | - | - | - | - |
| Max epipolar line [81] | R152 | 320 | 92.67 | 92.45 | 91.57 | 97.69 | 95.01 | 91.88 | - | - | - | - |
| Cross-View fusion * [38] | R50 | 320 | 95.6 | 95.0 | **93.7** | 96.6 | 95.5 | 92.8 | 96.7 | 96.5 | 96.2 | 95.9 |
| Cross-View fusion * [38] | R50 | 256 | 86.1 | 86.5 | 82.4 | 96.7 | 91.5 | 79.0 | **100** | 93.7 | 95.5 | 95.1 |
| Epipolar transformer [38] | R50 | 256 | 96.44 | 94.16 | 92.16 | 98.95 | **97.26** | 96.62 | 99.89 | 99.68 | 99.63 | 97.01 |
| **Mview-Joints (ours)** | R152 | 384 | **99.65** | **97.31** | **93.70** | **99.22** | 97.24 | **97.45** | 99.83 | **99.82** | **99.75** | **98.22** |

Table 4.2: Comparison of average median Euclidean distance error between Mview-Joints and proposed Learning triangulation backbone on Human3.6m. *Backbone*: Resnet 152 with pretrained weight [45].

| | Net | shlder | elb | wri | hip | knee | ankle | root | neck | nose | belly | head | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Learning triangulation [45] | Backbone | **7.84** | 8.00 | 7.40 | **7.55** | 7.45 | 9.70 | 5.75 | **5.86** | 6.46 | 6.47 | 6.57 | 7.18 |
| **Mview-Joints (ours)** | Backbone + Multi-view | 7.88 | **6.73** | **7.08** | 7.62 | **6.82** | **9.19** | **5.24** | 6.05 | **5.29** | **6.15** | **3.25** | **6.48** |


Ground truth

Results from Original Backbone

Results from Mview-Joints

Figure 4.4: Challenging poses, the multi-view scheme takes advantage from the additional view with respect to the backbone—single view—proposed by [45].

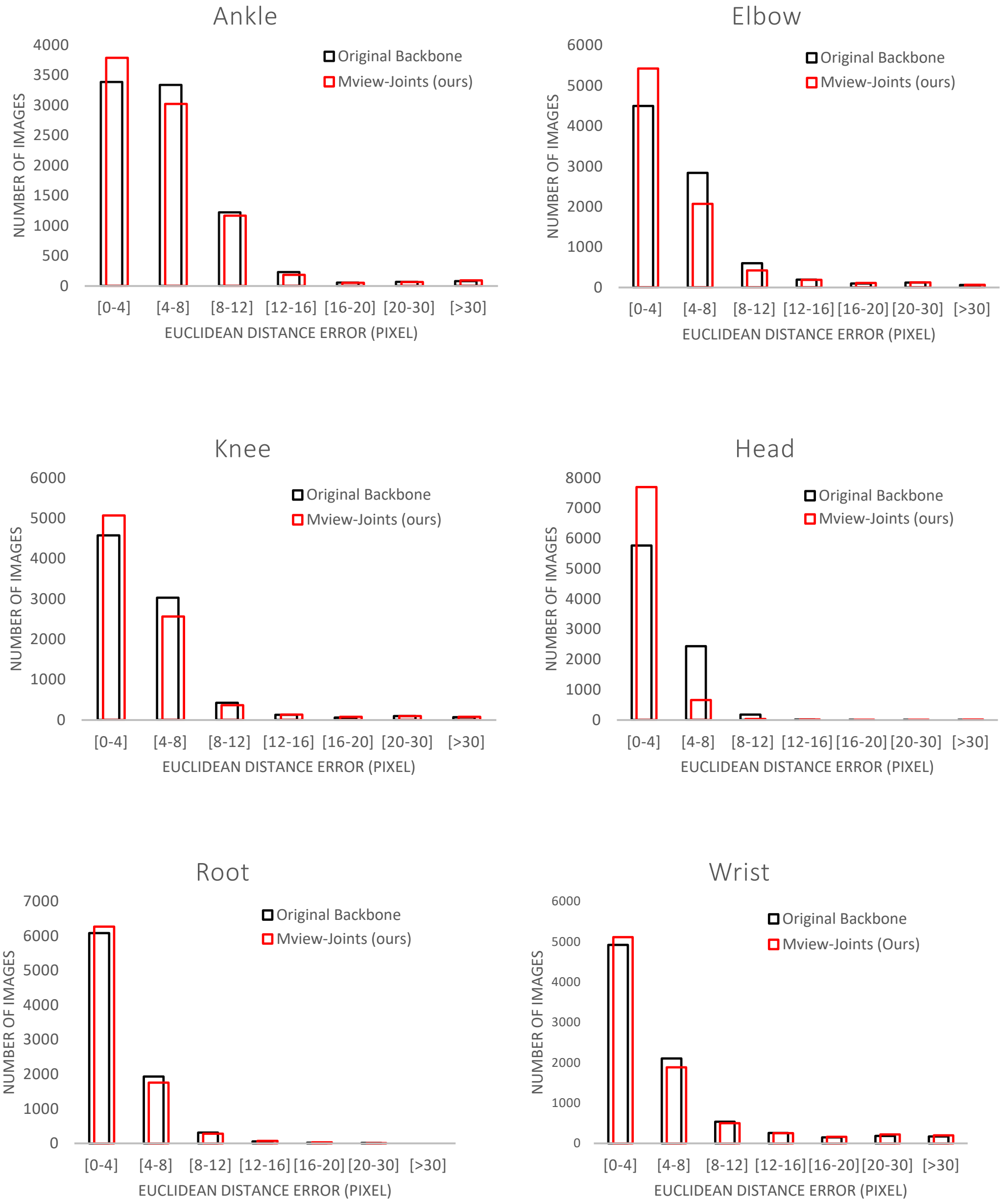


Figure 4.5: Comparison of Euclidean distance errors between the backbone approach [45] and the proposed Mview-Joints for six different body joints.

with two models, Cross-View fusion proposed by [81] together with its different variants and Epipolar transformer proposed by [38] by using the JDR metric. These state-of-the-art CNN-based methods were also trained considering a multi-view scheme. The details of models, with respect to the network architecture used as CNN backbone and the scale of images as input to the models, are denoted as "net" and "scale" respectively. The results for each body joints such as shoulder, elbow, wrist, hip, knee, among other, were obtained by using the mean of body joint on both sides of human body, i.e., left and right sides.

The comparisons depicted in Table 4.1 shows that Mview-Joints outperforms the previous works on most of body joints. The improvement is most significant for the shoulder, elbow, and ankle joints, which increment from 96.44% to 99.65%, from 95.00% to 97.31% and from 96.62% to 97.45%, respectively. The average JDR of body joints obtained by Mview-Joints improves the results of Epipolar transformer [38] about 1%, and with respect to Cross-View fusion [81] about 3% approximately.

Since the CNN backbone proposed by [45] is used as base into proposed approach, the median Euclidean distance error is used to evaluate the accuracy of prediction of the body joints. It is important to remember that JDR metric is used to indicate if a body joint is considered as a successful prediction but does not give any reference of its accuracy. The quantitative results are shown in Table 4.2. The elbow, wrist, knee, ankle, nose and head are the joints where an improvement in the accuracy is observed. An improvement in the media Euclidean distance errors of 15.88%, 4.32%, 8.46%, 5.25%, 18.11% and 50.53% respectively with respect to the results obtained with CNN backbone proposed by [45]. Figure 4.4 presents some challenging poses where Mview-Joints takes advantage of the different views with respect to the approach presented in [45]. In a quantitative analysis, both approaches present similar body joint accuracy, principally for the visible joints such as hip, head and root, since these joints are usually visible for most body pose. However, the accuracy obtained by Mview-Joints for the remainder of body joints are better than the predicted by [45]. Note that these body joints (i.e., shoulders and with them, elbow and wrist, which are part of upper limbs as well as knee and ankle, which are part of low limbs) have a high capacity of rotation/mobility, being more difficult to predict since they could be occluded due the natural body pose.

Figure 4.5 shows the histograms of accuracy obtained for each body joint for both approaches, i.e., the model presented by [45] and Mview-Joints. The histograms show that on average Mview-Joints obtain better estimation on the body joints, compared with the approach presented in [45]. Most of the predicted 2D positions of body joints are in the ranges [0-4] pixels and [4-8] pixels by using Euclidean distance error.

Additionally, the trained model is evaluated using other human pose datasets to determine the generality of this model to predict challenging poses on unknown indoor and outdoor environments. Figure 4.6 shows the obtained results of human pose estimation using Leed Sports Pose dataset, which contains images with different body poses in outdoor environment. The qualitative results show that the model is able to predict the body poses, although the images have been captured on unknown outdoor environment for the model. Likewise, Figures

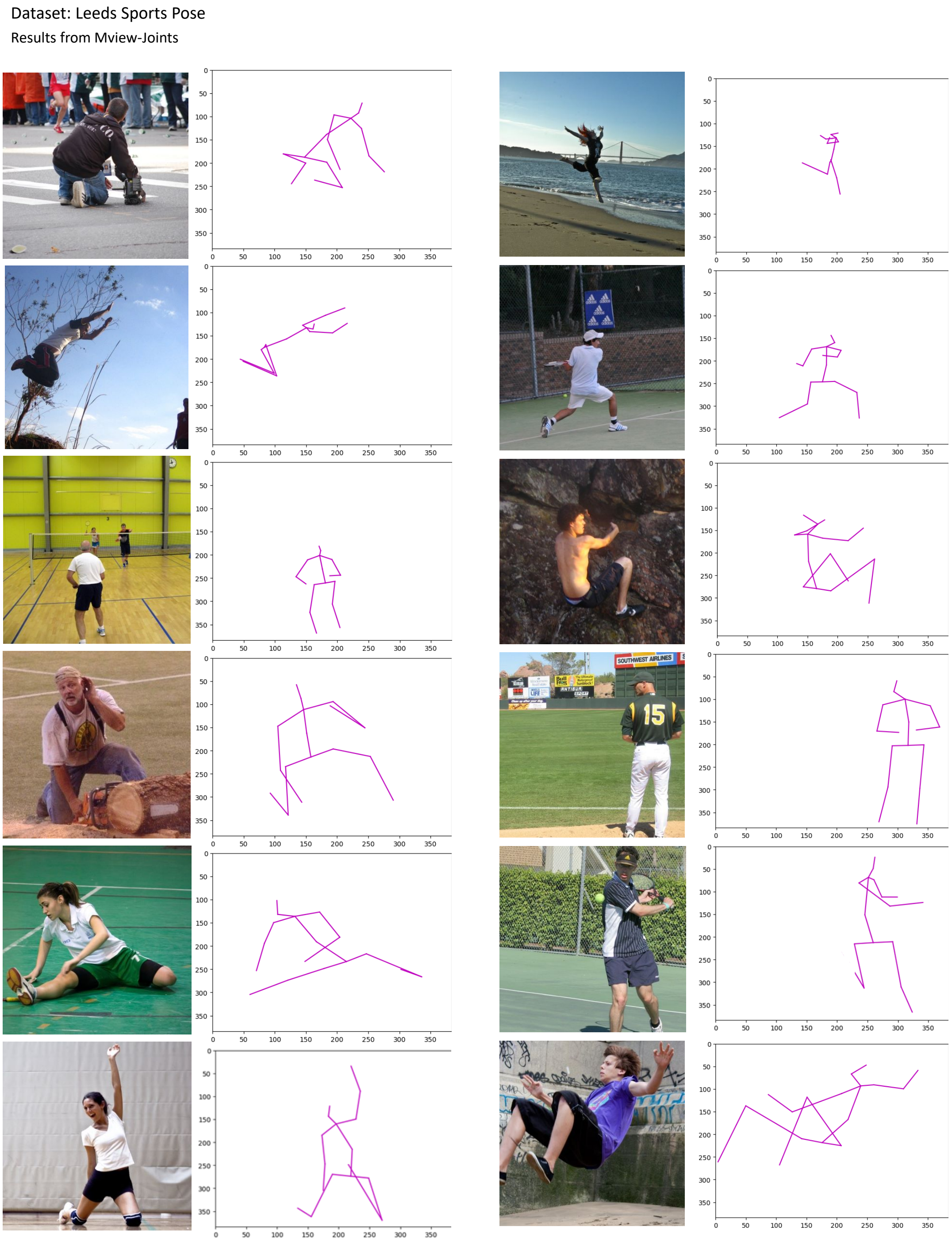


Figure 4.6: Leeds Sports Pose dataset [49] is used to evaluate the performance of the trained model called Mview-Joints on unknown outdoor environment.

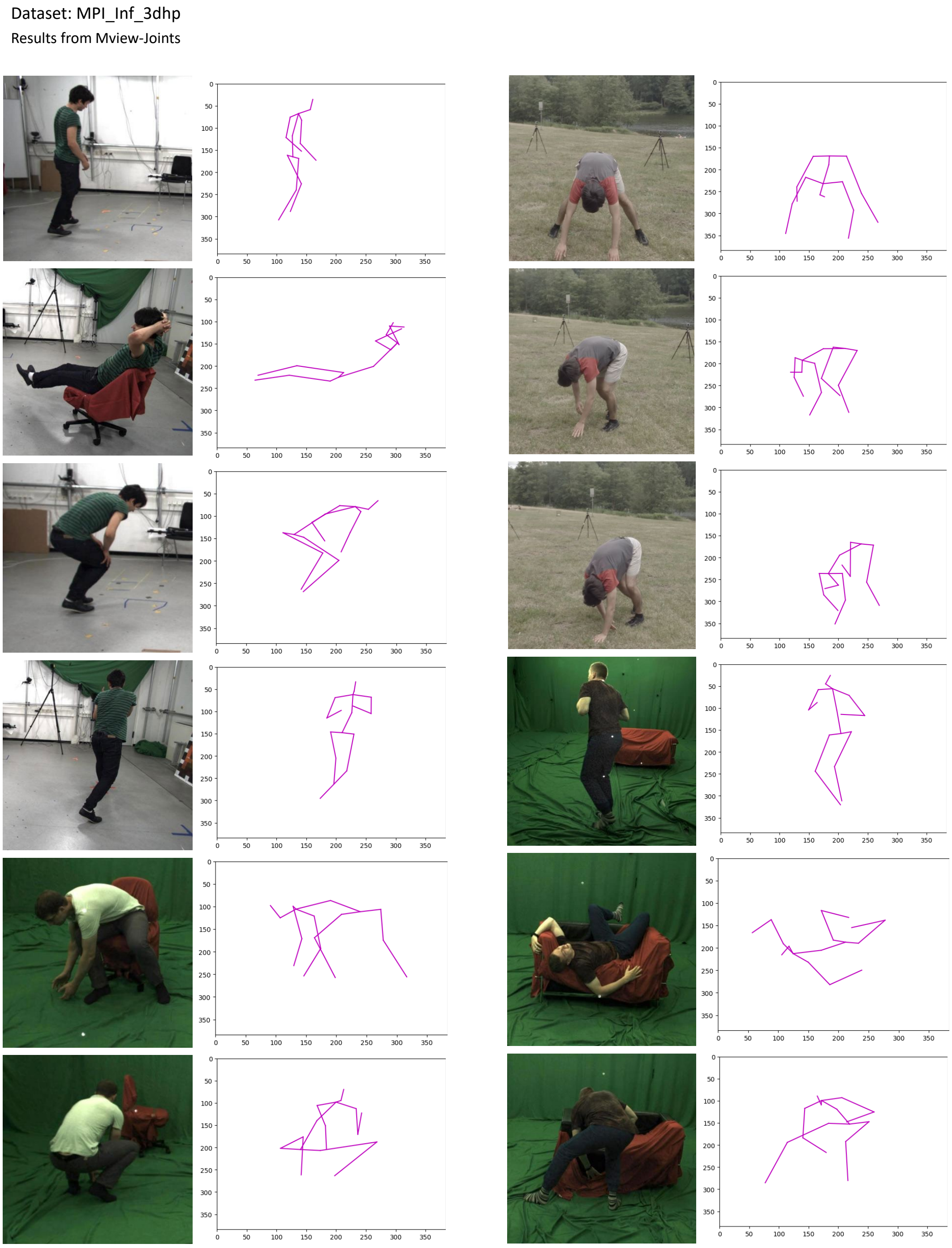


Figure 4.7: MPI-Inf-3DHP dataset [69] is used to evaluate the capacity of the trained model called Mview-Joints to predict challenging poses on unknown indoor and outdoor environments.

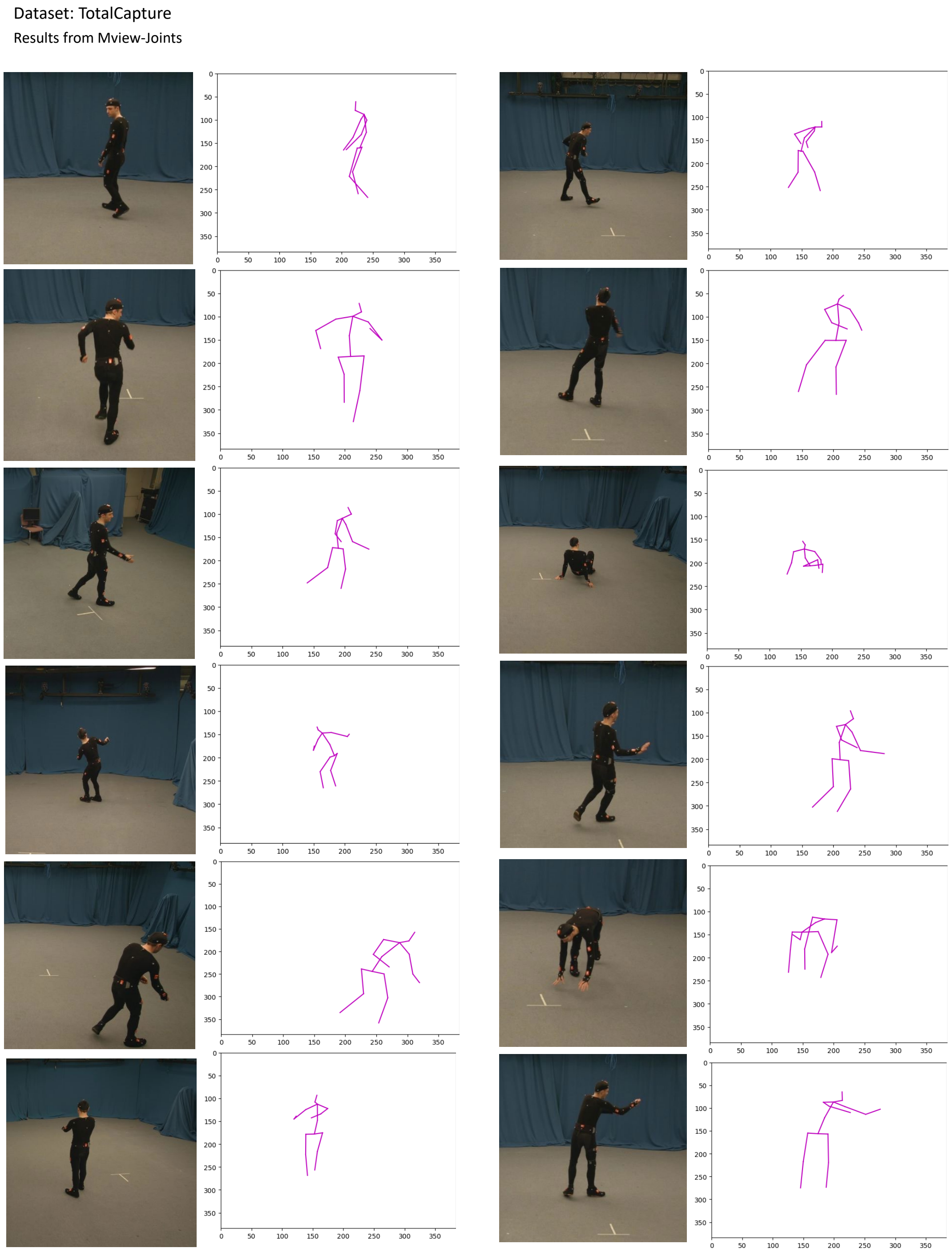


Figure 4.8: Total Capture dataset [113] is used to evaluate the capacity of the trained model called Mview-Joints to predict challenging poses on unknown indoor environment.

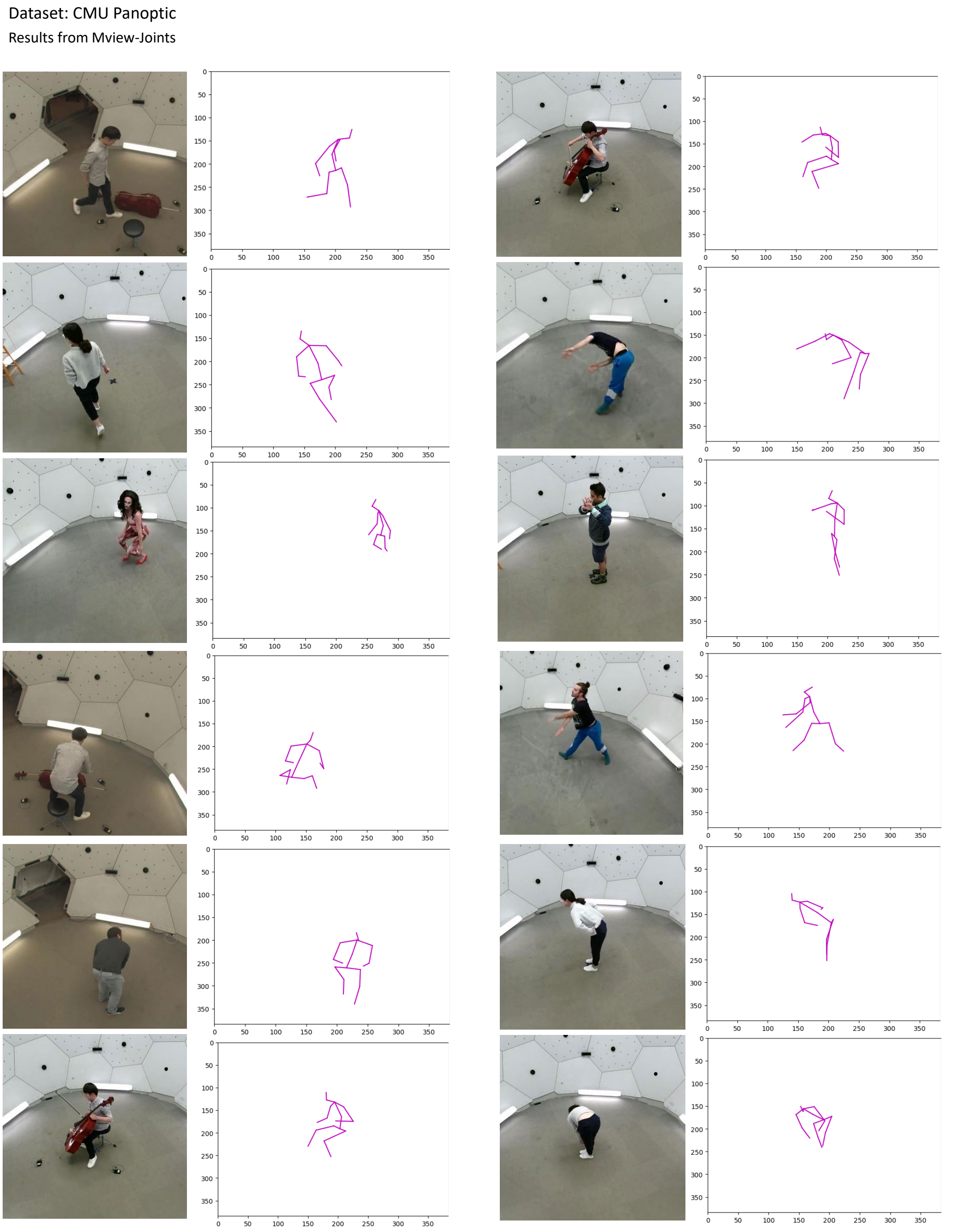


Figure 4.9: CMU Panoptic dataset [50] is used to evaluate the capacity of the trained model called Mview-Joints to predict challenging poses from different viewpoints on unknown indoor environment.

4.7, 4.8 and 4.9 show the generality of trained model to predict challenging human pose from MPI-Inf-3DHP, Total Capture and CMU Panoptic datasets respectively, which contain images captured from indoor or outdoor environments. Similar to the previous results, the model is able to predict challenging human poses shown appealing qualitative results. It is important to take in account that the images used for evaluation have been captured from different viewpoints as well as different indoor and outdoor environments.

## 4.3 Attention mechanisms based architecture

In this section, a review of the state-of-the-art of the different visual attention mechanisms are presented. Additionally, a new architecture, which uses the visual attention on images to tackle the human pose estimation problem from multi-view environments, is proposed. In contrast of the approach proposed above, this new approach does not take into account the information geometry of the cameras.

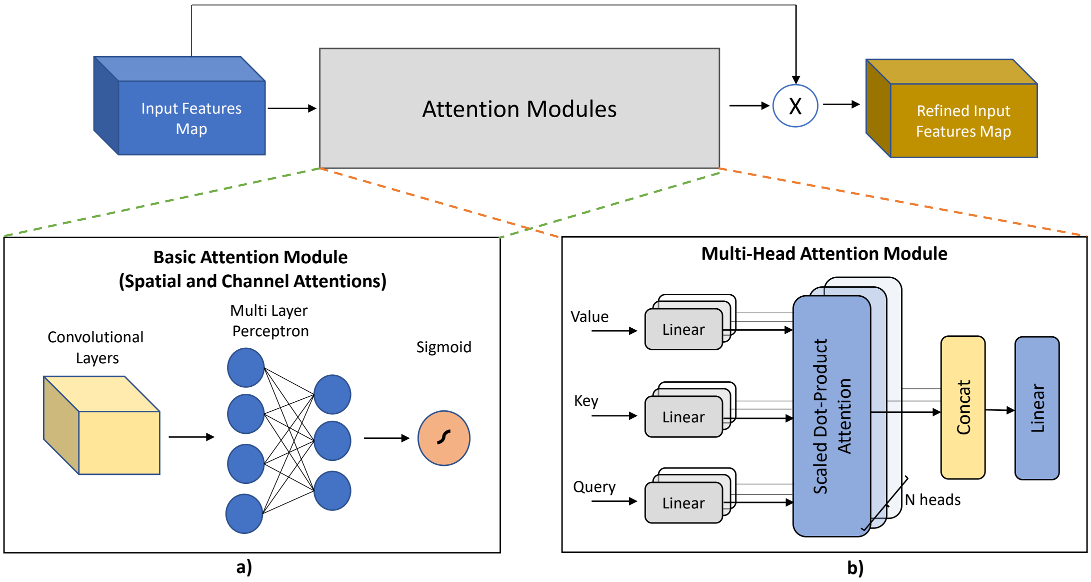


Figure 4.10: Base structure of attention modules: a) Channel and Spatial attention modules [126], b) Multi-Head attention module [115].

### 4.3.1 Attention Modules

Visual attentions are mechanisms that have allowed to identify important features while suppressing irrelevant information to understand the context of the scene. Attention mechanisms have been used to capture long-range features interactions to improve the performance of CNNs, since they allow to focus more on the important information instead of take attention to non-useful information, for instance, the background information in tasks such as object

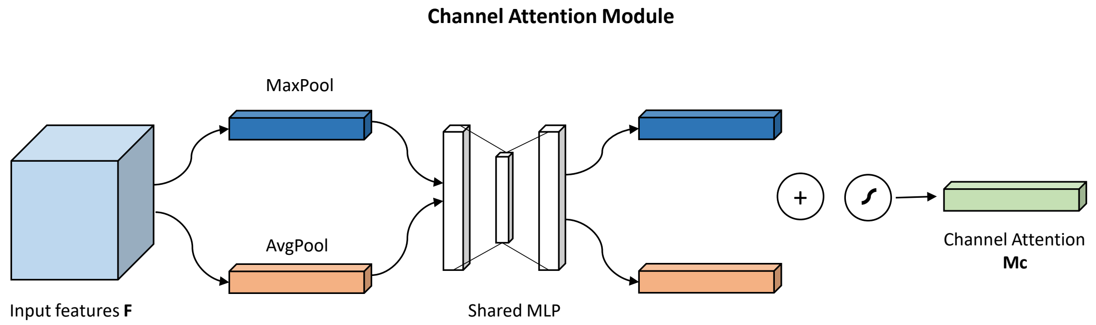


Figure 4.11: Structure of Channel Attention [126].

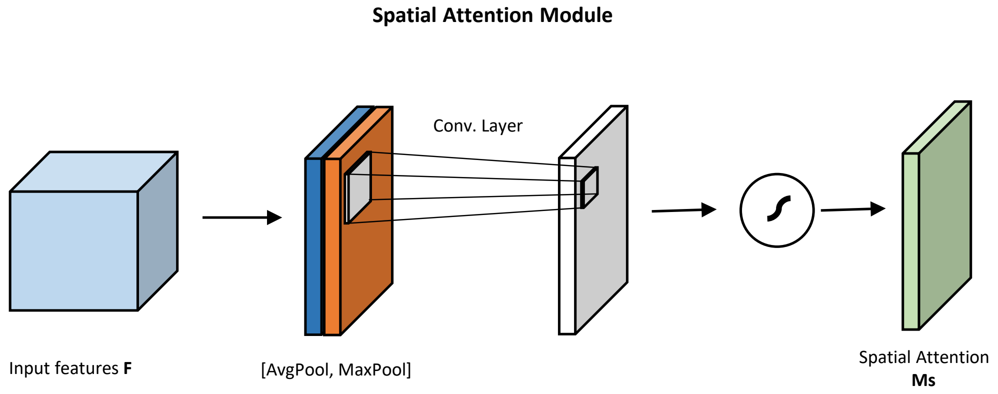


Figure 4.12: Structure of Spatial Attention [126].

detection, classification, body joints detection, among other. Although CNN architectures such as VGGNet [100], ResNet [37], GoogLeNet [106], Xception [16], just to mention a few, have demonstrated appealing results in various tasks in computer vision, the recent literature [33, 47, 133] shows that attention mechanisms can be used to improve the representation power of these features obtained from CNNs, i.e., they can extract important features by blending cross-channel and spatial information, and the Attention modules can learn ”where” and ”what” to focus to attend important features and suppress unnecessary ones of the activations.

Some Attention modules recently proposed are Channel Attention modules, Spatial Attention modules, Multi-Head Attention modules, just to mention a few. Generally, attention mechanisms are applied to spatial and channel dimensions, which consist of a 2D-Convolutional layer, Multi Layer Perceptron (in the case of channel attention), and Sigmoid function to generate a mask of the input features (see Fig. 4.10). The residual architecture was the first to experiment these attention mechanisms (channel and spatial attentions).

The attention maps of these channel attention modules are presented in [126], which are used to take full advantage of the inter-channel relationship of features. In details, each channel of these maps could be considered as a feature detector, which will allow focusing on 'what' is useful given an input image. Average-pooling and Max-pooling operations denoted as $F^c_{avg}$ and $F^c_{max}$ respectively, are used to get two different spatial context descriptors from input features. The channel attention map is obtained using both descriptors, which are sent to a shared network multi-layer perceptron (MLP) with one hidden layer. Finally, new descriptors are generated from the shared network, whose output feature vectors are merged using element-wise summation (see Fig. 4.11). The channel attention is formulated as:

$$M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))), \tag{4.9}$$

$$M_c(F) = \sigma(W_1(W_0(F^c_{avg})) + W_1(W_0(F^c_{max}))). \tag{4.10}$$

where $\sigma$ represents the sigmoid function, $W_0 \in \mathbb{R}^{C/r \times C}$, and $W_1 \in \mathbb{R}^{C \times C/r}$. Note that the MLP weights, $W_0$ and $W_1$, are shared for both descriptors.

Similarly to the architecture mentioned above, the same authors in [126] also propose an approach where the Spatial Attention maps are generated taking in account the inter-spatial relationship of features, which is focused in part of the information of input features, being complementary to the channel attention. The spatial attention is computed through average-pooling and max-pooling operations along the channel axis as $F^s_{avg} \in \mathbb{R}^{1 \times H \times W}$ and $F^s_{max} \in \mathbb{R}^{1 \times H \times W}$ respectively, which are concatenated to generate feature descriptors. Then, a convolution layer is applied to create a spatial attention map, which allows to consider important features and suppress unnecessary ones (see Fig. 4.12). The spatial attention is formulated as:

$$M_s(F) = \sigma(f^{7 \times 7}([AvgPool(F); MaxPool(F)])), \tag{4.11}$$

$$M_s(F) = \sigma(f^{7 \times 7}([F^s_{avg}; F^s_{max}])). \tag{4.12}$$

where $\sigma$ represents the sigmoid function and $f^{7x7}$ represent a convolution operation with the filter size of 7x7.

In [115], an attention mechanisms known as Multi-Head attention is proposed. Their attention mechanisms are run several time in parallel to capture relevant information from input (see Fig. 4.13), whose outputs are concatenated and linearly transformed. This attention mechanism could allow for attending longer-term dependencies (i.e., when the gap between the relevant features and the analyzed feature are very large) instead of shorter-term dependencies. A set of *"queries"* and *"keys"* with a vector of dimension $d_k$, and *"values"* with a vector of dimension $d_v$, are used. The Multi-head attention implements a scaled dot-product attention, which computes a *dot product* for each *"query"* with all *"keys"*. Each result is divided by $\sqrt{d_k}$, and then, a softmax is applied to get the weights on the *"values"*.

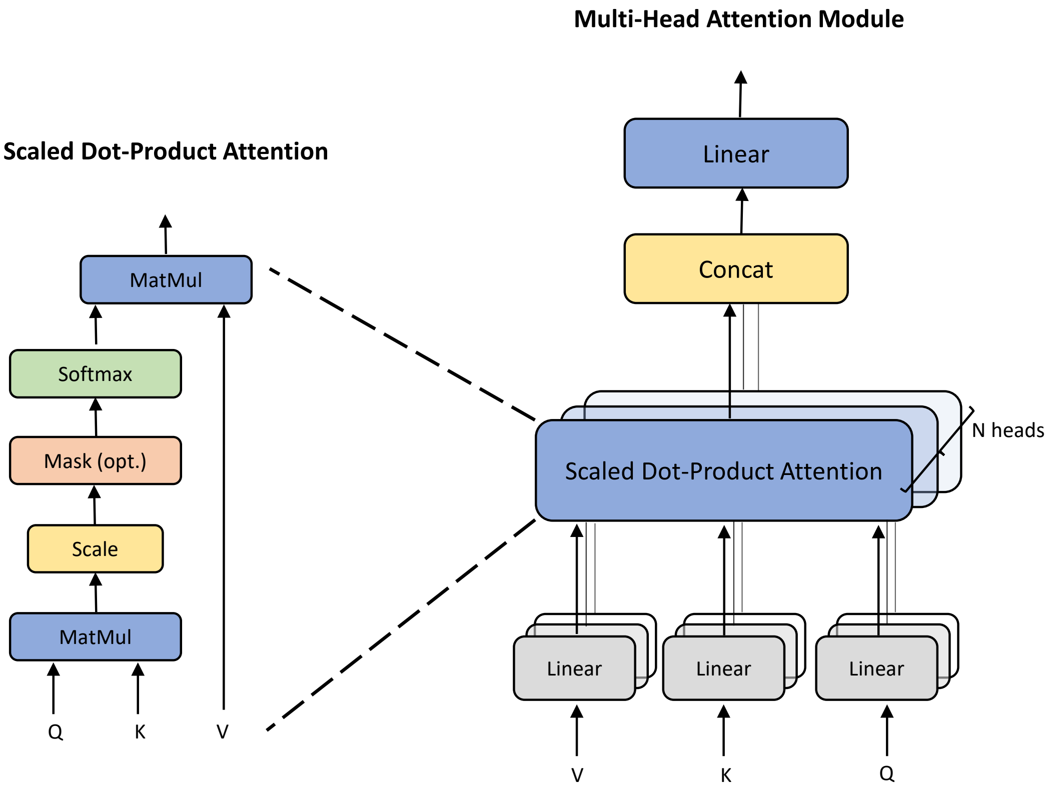

Scaled Dot-Product Attention
MatMul
Softmax
Mask (opt.)
Scale
MatMul
Q
K
V
Multi-Head Attention Module
Linear
Concat
Scaled Dot-Product Attention
N heads
Linear
Linear
Linear
V
K
Q

Figure 4.13: *(left)* Details of Scaled Dot-Product Attention. *(right)* Structure of Multi-Head attention [115].

In practice, the attention function is applied on a set of *"queries"* simultaneously, which are packed in the matrix $Q$. The *"keys"* and *"values"* are also packed together into matrices $K$ and $V$ respectively. According to [115], the matrix of outputs is computed as:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \times V. \tag{4.13}$$

The attention mechanism mentioned above is run $h$ times, i.e., linearly project $h$ times the *"queries"*, *"keys"* and *"values"* in parallel. These executions in parallel generate $h$ outputs, which, in turn, are concatenated and projected again to produce a final result. This has allowed to focus on relevant information from different representation subspaces at different positions. The multi-head attention is formulated as:

$$\begin{aligned} MultiHead(Q,K,V) &= Concat(head_1,...,head_h)W^o, \\ where\ \ head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V). \end{aligned} \tag{4.14}$$

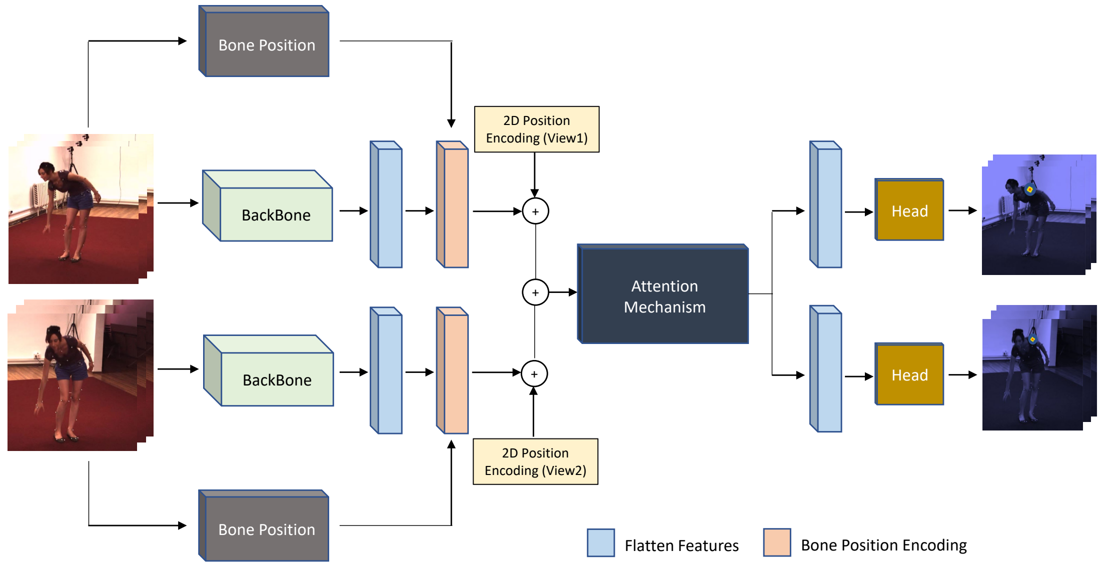


Figure 4.14: Overview of proposed architecture using an attention mechanism. It is fed with 2D position encoding of obtained features, including bone position encoding to guide the architecture to be more precise.

where $h$ corresponds to the number of attention mechanisms to execute in parallel and $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ and $W^o \in \mathbb{R}^{hd_v \times d_{model}}$ are parameter matrices.

Considering the advantages of attention mechanisms, they could be used to focus on relevant features of the body pose in the image, and thus, tackle the human pose estimation problem.

### 4.3.2 Network Architecture

This section presents the architecture proposed to leverage on attention mechanisms, which allow to capture relevant features in the image to tackle the task of single human pose estimation. The proposal consists on getting low-level image features from a CNN backbone; then an attention mechanism is used to capture long-range spatial relationship between these captured features; and finally, a head to predict the heatmaps of joints (see Fig. 4.14). According to [134], the initial part of the Resnet-50, i.e., just 5.5% of the 25.6M parameters available of original architecture, is considered as CNN backbone ß(.). In details, given a set of pairs of images $I_i \in \mathbb{R}^{3 \times H_i \times W_i}$, where $i \in 1,2$ represents view 1 and view 2. The proposed backbone ß(.) obtains the low-levels features, as shown below:

$$X_i = ß(I_i) \in \mathbb{R}^{d \times H \times W}, \tag{4.15}$$

where the number of channels is denoted as $d$, and the height and width of the feature map are $H$ and $W$, respectively.

The output of CNN backbone $X_i$, which is represented as 2D spatial structure image features, is flattened to generate a sequence vector $X_i' \in \mathbb{R}^{L\times d}$, where $L = H \times W$. The position information of generated sequence vector is encoded using a 2D sine positional encoding $E_{sin_i}$, including 2D bone positional encoding $E_{bpos_i}$, which are added onto $X_i'$, as shown below:

$$X = [X_i' + E_{sin_i} + E_{bpos_i}] \in \mathbb{R}^{nL\times d}, \tag{4.16}$$

where $n$ represents the number of views used for the proposed architecture. Finally, the standard attention mechanism $\xi(.)$, is fed with an uniform embedding (see Fig. 4.14). It is built by concatenating two views (i.e., $X_1'$ and $X_2'$) and computed using Eq. 4.16.

### Attention Mechanism

The attention mechanism $\xi(.)$ is composed of several multi-head self-attention, i.e., several layers that use linear projections of input to the attention mechanism obtained from Eq. 4.16, to generate a set of queries ($Q \in \mathbb{R}_{nL\times d}$), keys ($K \in \mathbb{R}_{nL\times d}$) and values ($V \in \mathbb{R}_{nL\times d}$). With them, three learnable matrices $W_q, W_k, W_v \in \mathbb{R}^{d\times d}$ are used to parameterize these linear projections. As shown Fig. 4.15, the scaled dot-product attention between $Q$ and $K$ is computed applying Eq. 4.13 to get the attention weights, and aggregate the values. This allows to capture how much dependency has each sequence by query-key-value attention. Finally, to compute the output of attention mechanism $\xi(X)$ denoted as $\hat{X}$, a multi-layer perceptron and skip connection are applied to the output of Eq. 4.13.

### Positional Encoding

Since the position and order of input sequence are necessary for the proposed attention mechanism, the positional encoding denoted as $E_{sin_i}$ is computed for each individual view following the Eq. 4.17 proposed by the original transformers in [115]. This allows to inject information about the relative or absolute position of the obtained features from CNN backbone. As these obtained features and positional encoding have the same dimension $d_{model}$, then both could be summed. The following equation shows the positional encoding corresponding to one view:

$$PE_{(pos,2m)} = sin(pos/10000^{2m/d_{model}}). \tag{4.17}$$

where $pos$ and $m$ correspond to the position information and its index respectively, and $d_{model}$ represents the dimension of feature vector.

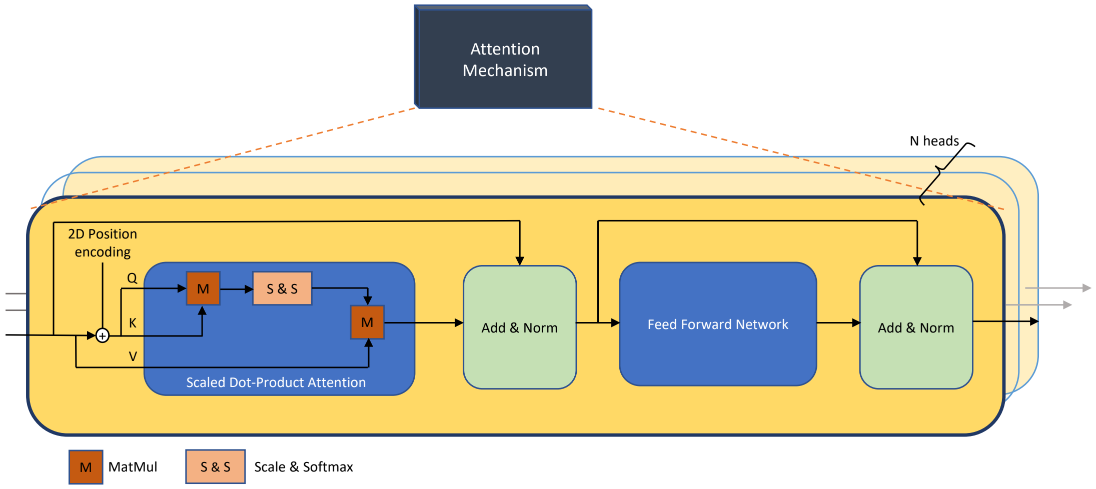


Figure 4.15: Details of attention mechanism [115] where 2D position encoding is considered to compute the attention weights from $Q$, $K$ and $V$. The obtained results are then applied to a Feed Forward Network to get the output of attention mechanism.

### Bone Position Encoding

A Bone position encoding denoted as $E_{bpos_i}$ is proposed to improve the attention mechanism on the image. In details, this new encoding contains information about position and orientation of body bones, mainly, those bones whose body joints have more probability to be occluded. For this, a person detector is considered to obtain the bounding box of persons, as shown below:

$$\beta_i = \delta_{detector}(I_i), \tag{4.18}$$

where $\beta$ corresponds to the bounding box of detected person, $\delta_{detector}(.)$ is the detector person available in the state-of-the-art, $I$ is the current image and $i$ corresponds to the $i$-*th* view (i.e., view 1 or view 2).

A CNN is implemented to get the contour lines of human pose from the input image. In details, the input image is transformed to a gray scale representation; then, a set of filters are defined, which are combined to set the weight of CNN. The output of CNN is mapped to other range of values, i.e., between 0 and 255.

Finally, a new function is built to generate a new image where a plain black background and the cropped contour-image are fused. This function places the cropped contour-image on plain black background in the same image coordinate system $(x, y)$ that original position of detected person in the input image; for this, the obtained coordinates of bounding box are used, which is formulated as:

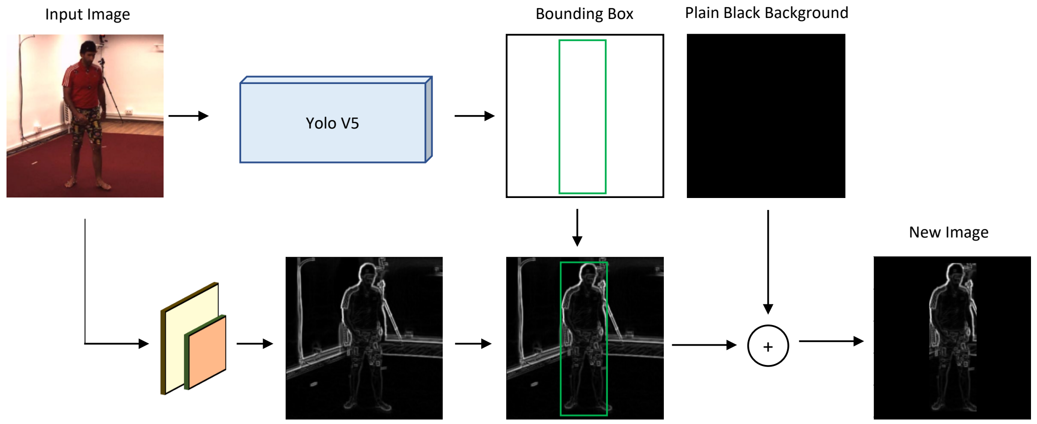


Figure 4.16: Overview of general process to get a new image from input image after obtaining the bounding box and contours.

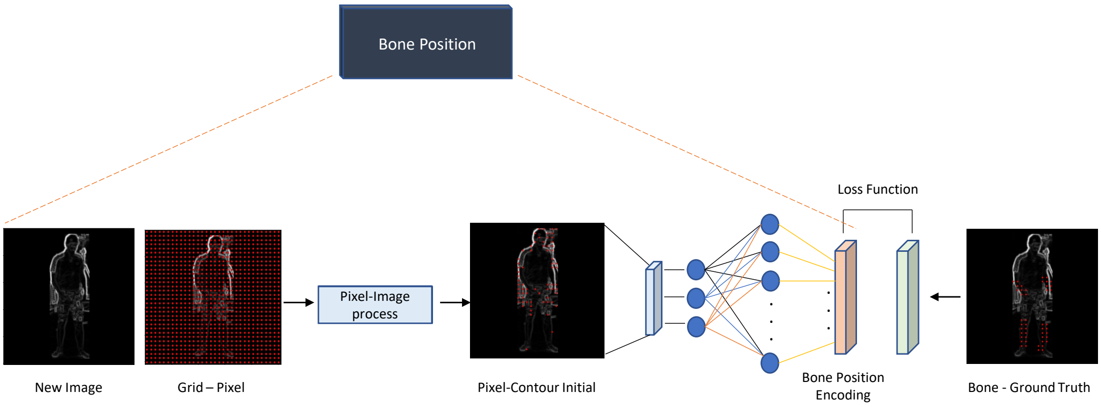


Figure 4.17: Details of an attention mechanism where an initial contour of a person is used to feed the neural network to obtain the *bone position encoding*.

$$\begin{aligned} Ic_i &= \triangle_{crop}(\beta_i(I_i)) \in \mathbb{R}^{h_1,w_1}, \\ NI_i &= \triangle_{image}(\delta_{\square}(.) \in \mathbb{R}^{h,w}, Ic_i \in \mathbb{R}^{h_1,w_1}), \end{aligned} \tag{4.19}$$

where $\triangle_{crop}(.)$ is a function to crop the contour-image according to the bounding box previously obtained, and $h_1$ and $w_1$ are the new height and width of the contour-image after cropping, $\delta_{\square}(.)$ is a function to build a plain black background with original height $h$ and weight $w$ of input image, and $\triangle_{image}(.)$ is a function that allows to fuse the obtained plain black background and cropped contour-image $Ic_i$ to get a new image (see Fig 4.16).

In order to identify the importance level of each pixel on new image, a new grid denoted as $G_{h_2 \times w_2}$ is built, where $h_2$ and $w_2$ are set a 32, and correspond to number of row and column of the grid. In details, the intensity value of pixel in the new image is obtained using the position that contains each cell of the grid. If the intensity values are near to 255, then the pixels are considered as important since they could be part of the contour lines of the body pose, and when these values are near to 0, then the pixels are considered as irrelevant. As a result of this process, a vector denoted as $V_{(h_2 \times w_2),d}$ where $d$ corresponds to the information of each row and column position of each pixel evaluated in the new image, and its intensity value respective. This vector, with information about the importance of each pixel, is used as input to the neural network to learn to map these features to ground truth of relevant pixel that make up the bones of body joints more complex, which is formulated as:

$$\begin{aligned} V_{ecd_i} &= \lambda_{\square}(G_{h_2 \times w_2}(NI_i)) \in \mathbb{R}^{(h_2 \times w_2),d}, \\ \varphi_i &= MLP(V_{ecd_i}) \in \mathbb{R}^{(h_2 \times w_2),d}, \end{aligned} \tag{4.20}$$

where $\lambda_{\square}(.)$ is a function that allows to identify the importance of each pixel in the new image previously obtained in Eq. 4.19, considering the usage of the new matrix $G_{h_2 \times w_2}$ detailed above, and whose output is denoted as $V_{ecd_i}$. The output of the new image of the *i-th* view is denoted as $\varphi_i$, which corresponds to the bone position encoding after applying a multi-layer perceptron to the information obtained in $V_{ecd_i}$.

### Head

Given the output of Attention Module denoted as $\tilde{X} = \xi(X) \in \mathbb{R}^{(h_2 \times w_2),d}$, a head is built to predict $K$ keypoints heatmaps of each view. For this, $\tilde{X}$ is split into $\tilde{X}_1$ and $\tilde{X}_2$, and then, a reshaping is performed to $\tilde{X}_i \in \mathbb{R}^{K \times H^* \times W^*}$, where $i$ represents each view available for the architecture, and $H^*$ and $W^*$ correspond to $H_i/4$ and $W_i/4$ respectively.

The prediction head denoted as $\Omega(.)$ is applied to reduce the channel dimension of $\tilde{X} = \xi(X)$ from $d$ to $K$, where one deconvolution and $1 \times 1$ convolution layer are used. In case that $H_i$ and $W_i$ are not equals, then an additional bilinear interpolation or a $4 \times 4$ transposed convolution could be considered. The equation is formulated as:

$$\Omega_i = H_{\square}(\perp(\tilde{X})) \in \mathbb{R}^{K \times H^* \times W^*}, \tag{4.21}$$

where $\perp(.)$ corresponds to split the output of the attention module for each view, and $H_{\square}(.)$ is a function that allows to get the heatmaps of body joints from them, whose result are saved in $\Omega_i$.

### Loss Function

In order to minimize the error during the learning process of relevant pixels between the bone position encoding generated for the images of *i-th* view and its ground truth, a loss function of bone position encoding is defined as:

$$Loss_{bpe} = \frac{1}{L \times d} \sum_{i=1}^{N} \left\| \varphi_i - \hat{p}_i \right\|_2, \tag{4.22}$$

where $L$ represents a matrix $(h_2 \times w_2)$, $N$ corresponds to the number of images, $\varphi_i$ corresponds to the bone position encoding of images of $i$-th view obtained from Eq. 4.20, and $\hat{p}_i$ is the ground-truth of features of the bone in the images of $i$-th view.

Finally, in order to get the general error of learning process of the architecture, the Mean Square Error (MSE) loss is applied, which is computed between the outputs of Head denoted as $\Omega_i$ obtained from Eq. 4.21 and the ground truth heatmap of 2D body joints of input images defined as 2D Gaussian centering around each keyjoint and denoted as $\hat{M}_i \in \mathbb{R}^{K \times H^* \times W^*}$. The equation used to train the architecture end-to-end is defined below:

$$Loss = \frac{1}{H^* \times W^*} \sum_{i=1}^{N} \left\| \Omega_i - \hat{M}_i \right\|_2 + Loss_{bpe}. \tag{4.23}$$

### 4.3.3 Experimental Results

Similarly to mentioned in Section 4.2.2, the experiments are conducted through of one large-scale pose estimation public dataset, including metrics to evaluate the performance of the proposed model. This section describes both of them, dataset and used metrics, including the obtained results.

#### Dataset

Human3.6m dataset is one of the largest publicly available human pose estimation dataset. It is generally used for multi-view setups since it uses four synchronized and calibrated digital cameras to capture all scene from different points of view. The persons perform different actions, which are used during the training and testing process of proposed model. The scheme used for these validations are the same as the one mentioned in Section 4.2.2.

#### Metrics

In order to evaluate the performance of the proposed model, the metrics mentioned in Section 4.2.2 are used. Thus, Joint Detection Rate (JDR) metric is used to measure the percentage of $successfully$ detected joints, considering a threshold. The Euclidean distance is also computed to estimated the accuracy of estimated joints in term of distance error.

### Training

Since the training process of the proposed architecture is oriented to use multi-view pose datasets, and they are quite limited, making very difficult to train it from scratch, the weights of pretrained Transpose proposed by [134], using MS-COCO dataset [60], are used to initialize the proposed architecture and finetune it on the Human3.6m dataset. The architecture is implemented using Pytorch and trained with NVIDIA Titan XP GPU and Intel Core I9 3.3GHz CPU. Following the settings in [38], Adam optimizer is used to train the network, including a learning rate of $10^{-3}$ and decays at 10-th and 15-th epoch with ratio 0.1. A back size of 16 (i.e., eight human poses simultaneously captured from two different points of view) is used for training model. As previous step, a pre-processing step is performed over the given dataset; it consists of cropping all images according to the bounding box, which has all the four sides of equal length with the person in the center for keeping the aspect ratio, and then, resizes them to 256×256 pixels. Additionally, the ground-truth of the relevant pixels of certain body parts are estimated for learning process of bone encoding position of each image input.

For the training process, a set of 156k images is used to feed to the network, which is trained until 20 epochs; it takes about 144 hours. The pre-processing mentioned above has been also used during the evaluation phase. In the evaluation a set of 8k images has been considered.

### Results and Comparisons

Quantitative results for human pose estimation using Attention Modules are depicted in Table 4.3. The evaluations of proposed architecture, referred to as Cross view Feature Bone, are compared with three models, Mview-Joints proposed by [9], Transpose proposed by [134] and Cross-View fusion proposed by [81], including their different variants by using the JDR metric. Cross view fusion and their variants (sum and max epipolar line) models are trained considering information of other views through of epipolar line for enhancing the performance of models. The details of models such as network architecture, scale of images and trainable parameters are denoted as "net", "scale" and "param" respectively. The results for each body joints such as shoulder, elbow, wrist, hip, knee, among others, were obtained by using the mean of body joint on both sides of human body, i.e., left and right sides.

The quantitative results depicted in Table 4.4 shows that Cross view Feature Bone using Attention Module outperforms the previous work on most of body joints, especially those whose network (backbone) and scale of images are the same than those used in the proposed architecture. The improvement is most significant if the model is compared with Cross-View fusion proposed by [81] and retrained by [38] using images of size (256x256 pixels), where body joints such as shoulder, elbow, wrist, knee and ankle have an increment from 86.1% to 95.4%, from 86.5% to 92.2%, from 82.4% to 88.8%, from 91.5% to 96.7% and from 79.0% to 91.0%, respectively, even when the scale of images used in Cross-view fusion model increases to 320x320 pixels, the proposed architecture has a slight advantage. This slight advantage is similar when it is compared with Transpose architecture proposed by [134]. When it is compared with Mview-Joints proposed in Section 4.2.1, the result is about 3% worse

Table 4.3: Comparison of 2D pose estimation accuracy on Human3.6m dataset using JDR(%) as metric. "−": these entries were absent. ∗ trained again by [38]. R50 and R152 are ResNet-50 and ResNet-152 respectively. Scale is the input resolution of the network. Param corresponds to the number of trainable parameters of models. AM means attention module. RCP corresponds to the relative camera pose.

| | net | scale | param | shlder | elb | wri | hip | knee | ankle | root | head | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sum epipolar line [81] | R152 | 320 | - | 91.36 | 91.23 | 89.63 | 96.19 | 94.14 | 90.38 | - | - | - |
| Max epipolar line [81] | R152 | 320 | - | 92.67 | 92.45 | 91.57 | 97.69 | 95.01 | 91.88 | - | - | - |
| Mview-Joints with RCP [9] | R152 | 384 | 80M | 99.65 | 97.31 | 93.70 | 99.22 | 97.24 | 97.45 | 99.83 | 99.75 | 98.02 |
| Transpose with AM [134] | R50 | 256 | 5M | 95.2 | 92.2 | 88.4 | 98.8 | 96.9 | 91 | 100 | 99.5 | 95.25 |
| Cross-View fusion * [38] | R50 | 320 | 525M | 95.6 | 95.0 | 93.7 | 96.6 | 95.5 | 92.8 | 96.7 | 96.2 | 95.26 |
| Cross-View fusion * [38] | R50 | 256 | 235M | 86.1 | 86.5 | 82.4 | 96.7 | 91.5 | 79.0 | 100 | 95.5 | 89.71 |
| Cross view Feature Bone with AM | R50 | 256 | 5M | 95.4 | 92.2 | 88.8 | 98.5 | 96.7 | 91 | 100 | 99.6 | 95.28 |

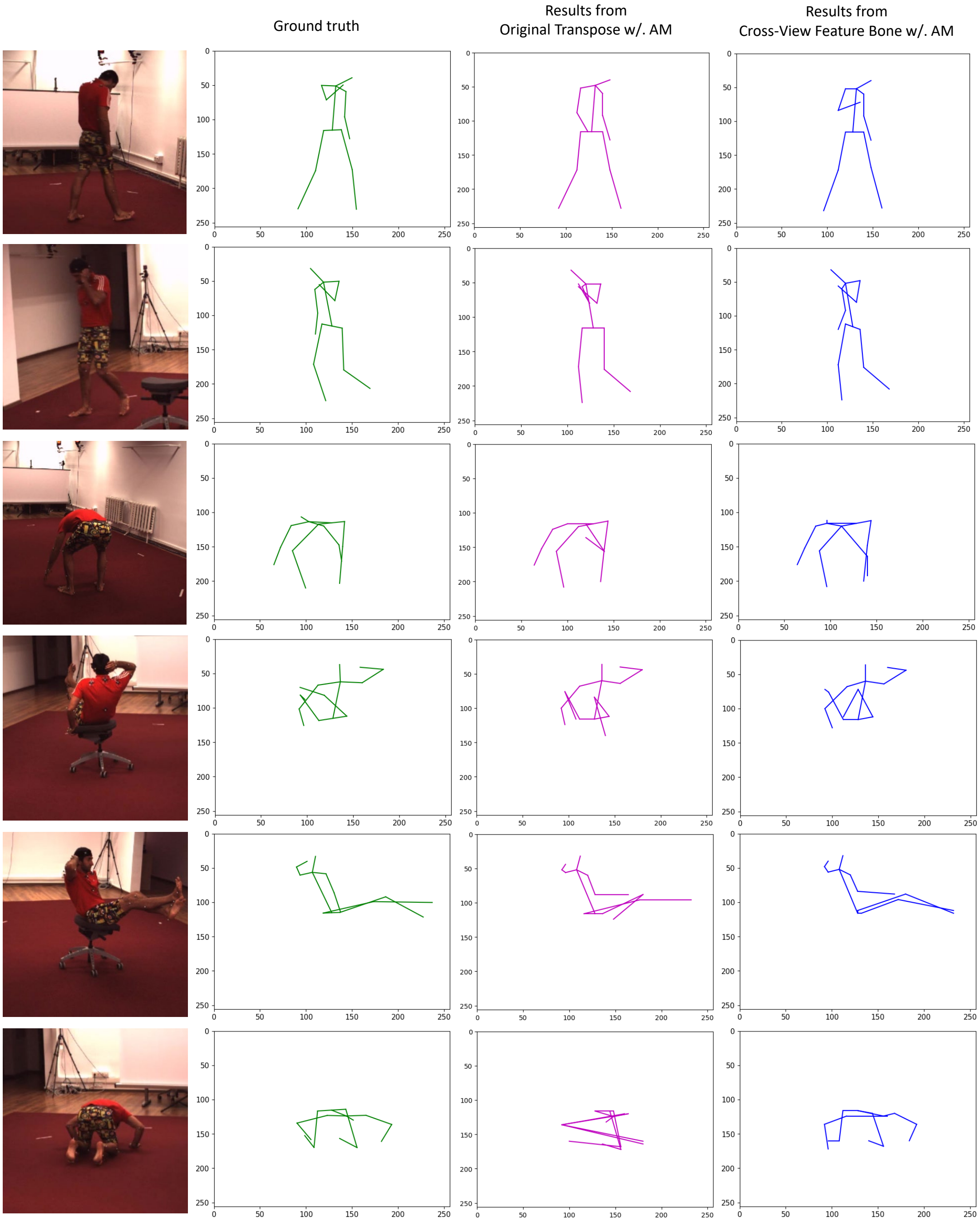


Figure 4.18: Challenging poses, Cross view Feature Bone using attention module takes advantage when the feature bone of one view is merged with feature body pose of other view with respect to Transpose architecture proposed by [134], which use a single view. AM means attention module.

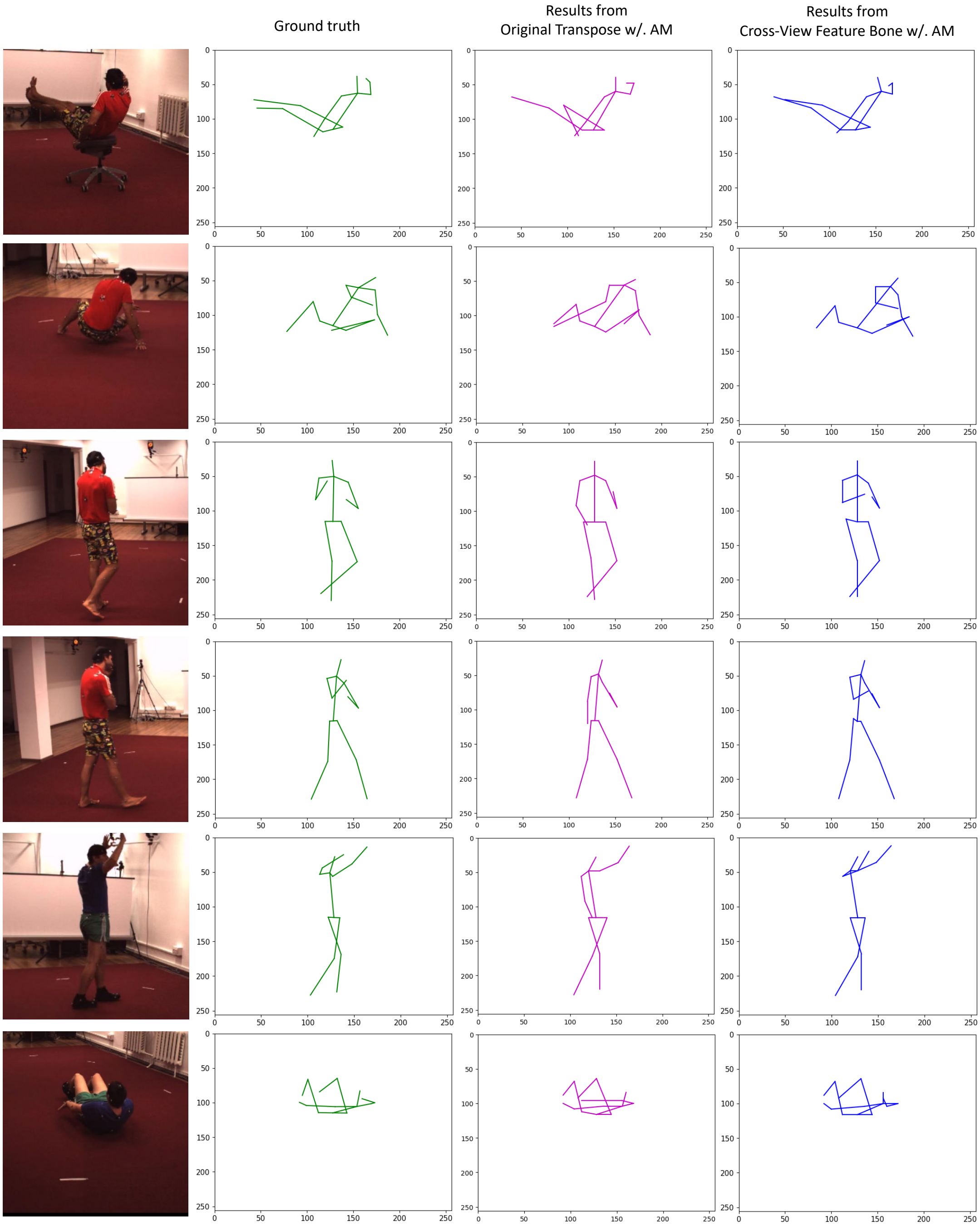


Figure 4.19: Other challenging poses where Cross view Feature Bone using attention module takes advantage with respect to the architecture proposed by [134]. AM means attention module.

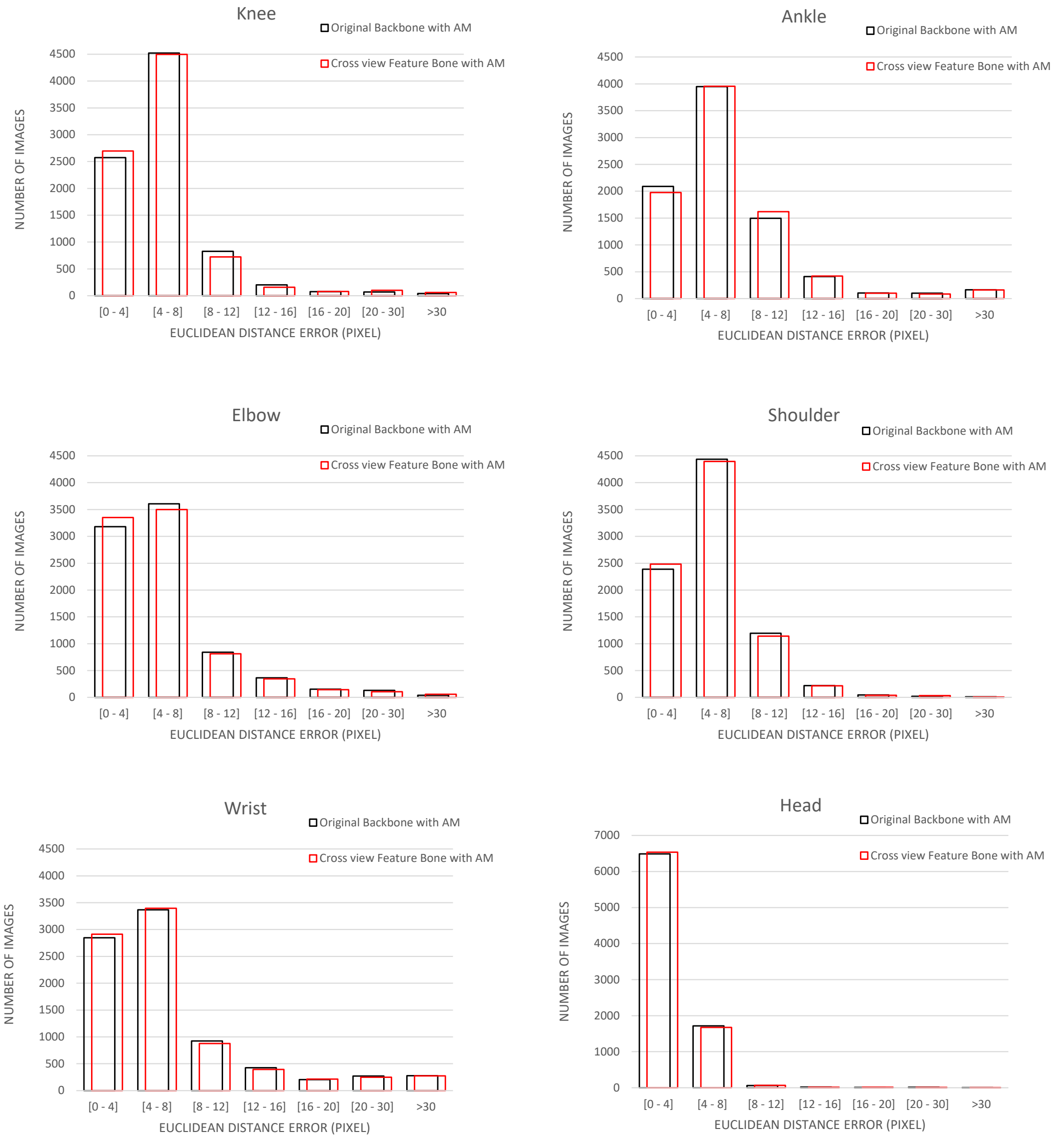


Figure 4.20: Comparison of Euclidean distance errors between the Transpose architecture proposed by [134] and the proposed Cross view Feature Bone with AM for six different body joints.

in average JDR of body joints, but if it is analyzed in details, the backbone used as network (Resnet 152), scale of images (384×384 pixels) and trainable parameters (80M) has a larger computational cost than Cross view Feature Bone. Likewise, the body joints with most impact about performance of model are elbow, wrist and ankle, which have major probability of having some type of occlusions and that are better solved by using relative camera pose as additional information in the learning process. On the other hand, the number of trainable parameters of the proposed architecture increments from 5M to 80M, 235M and 525M, depending on the backbone used as network and the scale of images, requiring large computing power.

Additionally, median Euclidean distance error is used to evaluate the accuracy of prediction of Cross view Feature Bone using Attention Module with respect to Transpose proposed by [134]. Note that JDR metric is used to determine if a body joint is considered as successful prediction taking into account a threshold. However, this metric does not give any reference of accuracy of each body joint.

Looking at the quantitative results presented in Table 4.4, it can be appreciated that an improvement in accuracy happens in shoulder, wrist, hip, knee. The median Euclidean distance errors for these joints are 1.18%, 2.58%, 0.28% and 1.57% with respect to the results obtained with Transpose architecture proposed by [134]. Figures 4.18 and 4.19 present some challenging poses where Cross view Feature Bone architecture takes advantage of information about position and orientation of bones in the image plane with respect to the approach proposed by [134]. In a quantitative analysis, both proposals present similar body joint accuracy, principally for the visible joints such as head and root, since these joints are usually visible for most body poses. However, the accuracy obtained by Cross view Feature Bone architecture for the remainder of body joints are better than the prediction from the model proposed by [134]; mainly body joints such as shoulder, wrist or knee that have a high capacity of rotation/mobility, being more difficult to predict since they could be occluded due the natural body poses.

The histograms of accuracy of each body joint for both proposed are shown in Fig. 4.20, i.e., the model presented by [134] and Cross view Feature Bone. The histograms show that Cross view Feature Bone obtained the body joints slightly more precise compared with the approach presented in [134]. Most of the predicted 2D positions of body joints are in the ranges [0-4] pixels and [4-8] pixels by using Euclidean distance error.

Additionally, Figure 4.21 shows the qualitative comparison of some challenging poses between architecture proposed by [9] called Mview-Joints with RCP and Cross View Feature Bone with Attention Module. In this qualitative analysis, Mview-Joints architecture with RCP has a slightly better performance than Cross View Feature Bone to predict the human body pose, mainly, for occluded joints. This could be because the Mview-Joints architecture with RCP has been trained using relative camera pose as part of its learning process.

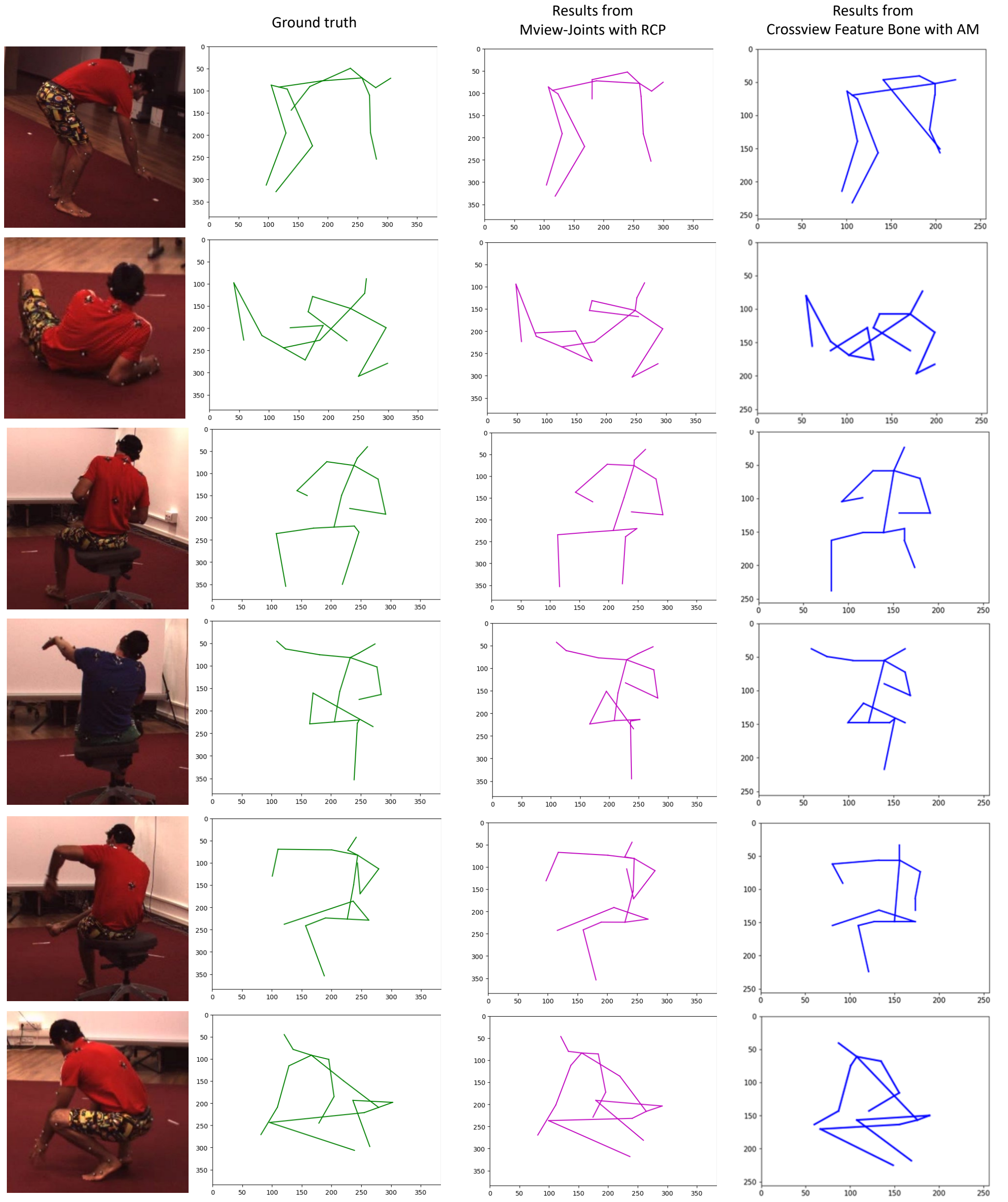


Figure 4.21: Comparison of qualitative results between Mview-Joint with RCP [9] and Crossview Feature Bone with attention module for different challenging poses.

Table 4.4: Comparison of average median Euclidean distance errors between Cross view Feature Bone with AM and Transpose with AM proposed by [134] on Human3.6m. AM means attention module.

| Model | shlder | elb | wri | hip | knee | ankle | root | head | Avg |
|---|---|---|---|---|---|---|---|---|---|
| Transpose with AM [134] | 5.10 | **4.35** | 4.65 | 3.59 | 4.45 | **5.61** | 1.24 | 2.90 | 3.99 |
| **Cross view Feature Bone with AM** | **5.04** | 4.40 | **4.53** | **3.58** | **4.38** | 5.74 | **1.24** | **2.90** | **3.97** |

## 4.4 Conclusions

This chapter focuses on answering two of the research questions proposed in this thesis, which are how the relative camera pose from multi-view environments could help to estimate occluded human body parts, and if the accuracy of the human body pose estimation from multi-view environments can be increased when the relative camera pose is considered. The proposed schemes are motivated by the reduced information to predict more precisely occluded joints when only one view is available to capture the scene. In order to tackle the multi-view scheme, the relative positions and orientations—extrinsic camera parameters—between the different cameras in the scene are considered. The usage of relative camera pose into learning process in the multi-view scheme has shown appealing results on challenging body poses with respect to state-of-the-art methods, increasing the accuracy of those joints that can be easily occluded due the natural body human pose such as shoulder, elbow, knee, wrist and ankle. This is possible since the body joints predictions together with their confidences obtained in each camera, can be transformed from a camera coordinate system in the source view to another in the reference view by using the relative camera pose, and then, project them to the image plane of reference view. The body joints predicted with better accuracy are fused with other body joints obtained from other camera coordinate system, whose predicted body joints are inaccurate since the camera is not able of capturing with accuracy the body joints occluded by the body pose. Likewise, the attention modules have allowed to design lightweight architectures to tackle the human pose estimation problem using the integration of relevant features on the images, specifically, position and orientation of bones in image plane from both self views and reference views. The usage of these attention modules has shown appealing results in spite of the low number of trainable parameters used for learning process respect to the trainable parameters used in models of state-of-art. The chapter shows how the information of body joints from other views can be fused to help the estimation of occluded joints more accurately than single view approaches, in particular when self-occlusions are considered. An important aspect to consider is that the accuracy of body joint estimations can help to solve other related problems, such as action recognition, surveillance, healthcare, 3D human pose estimation, among other.

# 5 Conclusions and Future Work

This chapter presents a summary of the whole research work. Additionally, the contributions that tackle the issues raised in this thesis and future works identified during these years are also described.

## 5.1 Conclusions

This dissertation presents works focused in the field of computer vision using images from multi-view camera systems and deep learning techniques through convolutional neural networks. Specially, different Siamese networks are proposed, which contain two or more identical subnetworks (i.e., same configuration with the same parameters and weights), being useful to find similarity/relationship between the images by comparing its feature vectors. Considering this type of neural network, different approaches have been proposed to solve challenging problems related with the camera pose and human pose estimation in multi-view environments.

Extensive literature review is presented in the Chapter 2, where different approaches proposed in the state-of-the-art to tackle the issues raised in this thesis are presented. The review of the state-of-the-art in the most relevant computer vision conferences and journals has allowed us to identify their advantages and limitations, helping us to formulate new approaches using deep learning techniques to solve the problems proposed in the thesis, and thus, contribute to the state-of-the-art.

In order to tackle the camera pose estimation problem from multi-view environments a novel architecture is proposed. It is based on Siamese network where the supervised learning is considered. The images used for the training are generated through of a multi-view system, which capture the scenario from different positions by different cameras at the same time. This approach has obtained better results than other approaches in the state-of-the-art on *Cambridge Landmarks* dataset. However, to overcome the limitation of having a large dataset of real-world images for the training process of the supervised scheme, a domain adaptation strategy has been proposed in this thesis. It consists of different virtual scenarios using special 3D simulation software (CARLA); on this virtual scenarios images are acquired according to the real-world scenarios, and thus, take advantage of transferring the learned knowledge from these virtual scenarios to real-world scenarios. Use a domain adaptation strategy improves the obtained results since that feature extracted on theses images of virtual scenarios help to have a better approximation of learned knowledge to adapt it to the real-world scenarios. However, it should be mentioned that the virtual environments' contents should have similar features with respect to the real environments' contents, including also the way of how are captured these images from virtual scenarios, i.e., distance and point of view between camera and objects.

For the second problem of the thesis, i.e., human pose estimation, the proposed approach uses the estimated extrinsic parameters to establish the relationship between all cameras into the multi-view scheme, where a supervised learning scheme is considered. For the training process, *Human3.6* dataset is used, whose images are generated from a multi-view system into a controlled environment. The cameras into the multi-view system capture the person from different positions at the same time, helping the proposed approach to leverage the redundancy of information generated from all cameras available in the scene. Additionally, the approach leverages the relative camera pose to establish their relationship between them, and thus, overcome the challenging problem of the human pose estimation when the joints are occluded due to the natural body pose. The correctly estimated joints from another view are mapped to the current view using the relative camera pose to improve those poorly estimated joints. It is motivated by the reduced information to predict more precisely occluded joints when only one view is used. The experimental results have shown that including the information of relative camera pose into the learning process of the proposed approach improves the obtained results. The main reason is the redundancy of feature information, which is generated by all cameras when the person is captured from a multi-view system. Furthermore, the capability of the proposed approach has allowed to enhance the poorly estimated joints since map the estimated joints correctly to other camera coordinate system using the relative camera pose. Additionally, another approach is proposed to tackle the human pose estimation problem; it uses attention modules. This approach does not take into account the relative camera pose, instead, the position and orientation of bones of human body are used as additional information, mainly, those bones of human body that could be occluded by the natural movements of body parts such as arms, forearms, legs among other. Appealing results have been obtained despite having fewer trainable parameters (5M of trainable parameters) than other approaches in the state-of-the-art (80M - 525M of trainable parameters). This lightweight architecture can be used on computer hardware and electronic devices that do not have high computing power. However, the estimation of body joints such as wrist, ankle and elbow, are not accurate if they are compared with the proposed approach that uses the relative camera pose as additional information. The main reason is that do not have information of another view of the occluded joints, making that the estimation of them is based only on position and orientation of bone of human body, which could also be occluded depending on the body pose.

### 5.1.1 List of Contributions

This thesis results on the following publications, in chronological order:

- **Jorge L. Charco**, Boris X. Vintimilla, and Angel D. Sappa. Deep learning based camera pose estimation in multi-view environment. In International Conference on Signal-Image Technology & Internet-Based Systems, pages 224–228, 2018.
- **Jorge L. Charco**, Angel D. Sappa., Boris X. Vintimilla., and Henry O. Velesaca. Transfer learning from synthetic data in the camera pose estimation problem. In International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, pages 498–505. INSTICC, SciTePress, 2020.
- **Jorge L. Charco**, Angel D. Sappa, Boris X. Vintimilla, and Henry O. Velesaca. Camera pose estimation in multi-view environments: From virtual scenarios to the real world. Image and Vision Computing, Vol.110:104182, 2021. **Journal**.
- **Jorge L. Charco**, Angel D Sappa, and Boris X Vintimilla. Human pose estimation through a novel multi-view scheme. In International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, pages 855–862. INSTICC, SciTePress, 2022.
- **Jorge L. Charco**, Angel D. Sappa, Boris X. Vintimilla, and Henry O. Velesaca. Human Body Pose Estimation in Multi-view Environments. ICT Applications for Smart Cities, pages 79-99. Springer, 2022.

## 5.2 Future Work

As a future work for camera pose estimation, other architectures based on recurrent neural networks that enable the model to be capable of take into account the temporal information of sequence of images will be considered. Likewise, architectures based on vision transformer (ViT), which use various small blocks of the image (same size block) as input to the transformer encoder, instead of using convolutional neural network that take as input each pixel of the image, will be also considered. Additionally, new loss functions will be implemented to improve the already obtained results.

Estimation of human body pose is crucial to tackle other problems related to the persons, such as action recognition, physical therapy or safe moving assistance for the elderly. The proposed approaches help to improve the accuracy of occluded body joints estimation through of multi-view schemes. Future approaches will address temporal information of the action performed by the person in the sequence of images. Likewise, the attention modules and vision transformer (ViT) will be considered together with the extrinsic parameters of each camera of the multi-view system. Since the datasets for human pose estimation from multi-view systems are generated in controlled environments, and to generate datasets in outdoor environments is complex due to the configuration of multi-view system, the usage of virtual scenarios will be considered. This will allow to generate almost an unlimited set of synthetic images considering different weather conditions and persons doing different actions in outdoor environments, which will be captured from different cameras at the same time. The new datasets will be used to train semi-supervised or unsupervised learning models, and thus, they can be compared with the results obtained using supervised learning.

# Bibliography

[1] Kassem Al Ismaeil. *Structure from Motion & Camera Self-Calibration.* PhD thesis, 06 2011.

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *International Conference on Computer Vision and Pattern Recognition*. IEEE, June 2014.

[3] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, Vol.110:346–359, 2008.

[4] Oleksandr Bogdan, Viktor Eckstein, Francois Rameau, and Jean-Charles Bazin. Deepcalib: a deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. In *European Conference on Visual Media Production*, pages 1–10, 2018.

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *International Conference on Computer Vision and Pattern Recognition*, pages 7291–7299. IEEE, 2017.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[7] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *International Conference on Computer Vision and Pattern Recognition*. IEEE, 2016.

[8] Lluís Castrejón, Yusuf Aytar, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Learning aligned cross-modal representations from weakly aligned data. In *International Conference on Computer Vision and Pattern Recognition*, pages 2940–2949, 2016.

[9] Jorge L Charco, Angel D Sappa, and Boris X Vintimilla. Human pose estimation through a novel multi-view scheme. In *International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 855–862. INSTICC, SciTePress, 2022.

[10] Jorge L. Charco., Angel D. Sappa., Boris X. Vintimilla., and Henry O. Velesaca. Transfer learning from synthetic data in the camera pose estimation problem. In *International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 498–505. INSTICC, SciTePress, 2020.

[11] Jorge L Charco, Angel D. Sappa, Boris X. Vintimilla, and Henry O Velesaca. Camera pose estimation in multi-view environments: From virtual scenarios to the real world. *Image and Vision Computing*, 110:104182, 2021.

[12] Jorge L. Charco, Boris X. Vintimilla, and Angel D. Sappa. Deep learning based camera pose estimation in multi-view environment. In *International Conference on Signal-Image Technology & Internet-Based Systems*, pages 224–228, 2018.

[13] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao. Pre-trained image processing transformer. In *International Conference on Computer Vision and Pattern Recognition*, pages 12294–12305. IEEE, 2021.

[14] Kefan Chen, Noah Snavely, and Ameesh Makadia. Wide-baseline relative camera pose estimation with directional learning. In *International Conference on Computer Vision and Pattern Recognition*, pages 3258–3268. IEEE, 2021.

[15] Davide Chicco. Siamese neural networks: An overview. *Artificial Neural Networks*, pages 73–94, 2021.

[16] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807. IEEE, 2017.

[17] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J. Black. Monocular expressive body regression through body-driven attention. In *European Conference Computer Vision*, page 20–40. Springer-Verlag, 2020.

[18] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units(elus). In *International Conference on Learning Representations*, 2016.

[19] Elliot J. Crowley and Andrew" Zisserman. In search of art". In *European Conference Computer Vision*, pages 54–70. Springer, 2015.

[20] Jian S. Dai. Euler–rodrigues formula variations, quaternion conjugation and intrinsic connections. *Mechanism and Machine Theory*, pages 144–152, 2015.

[21] Dong-Luong Dinh, Myeong-Jun Lim, Nguyen Duc Thang, Sungyoung Lee, and Tae-Seong Kim. Real-time 3d human pose recovery from a single depth image using principal direction analysis. *Applied Intelligence*, page 473–486, September 2014.

[22] Yongtae Do. Application of neural networks for stereo-camera calibration. In *International Conference on Neural Networks*, pages 2719–2722. IEEE, 1999.

[23] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[24] Marcin Eichner and Vittorio Ferrari. We are family: Joint pose estimation of multiple persons. In *European Conference on Computer Vision*, pages 228–242. Springer, 2010.

[25] Marcin Eichner, Vittorio Ferrari, and S Zurich. Better appearance models for pictorial structures. In *British Machine Vision Conference*, pages 1–11, 2009.

[26] Sovann En, Alexis Lechervy, and Frédéric Jurie. Rpnet: An end-to-end network for relative camera pose estimation. In *European Conference on Computer Vision*, pages 738–745. Springer, 2018.

[27] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *International Conference on Computer Vision*, pages 2334–2343. IEEE, 2017.

[28] Olivier D Faugeras, Q-T Luong, and Stephen J Maybank. Camera self-calibration: Theory and experiments. In *European Conference on Computer Vision*, pages 321–334. Springer, 1992.

[29] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient matching of pictorial structures. In *International Conference on Computer Vision and Pattern Recognition*, pages 66–73. IEEE, 2000.

[30] Martin A Fischler and Robert A Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on computers*, pages 67–92, 1973.

[31] Rawia Frikha, Ridha Ejbali, and Mourad Zaied. Camera pose estimation for augmented reality in a small indoor dynamic scene. *Journal of Electronic Imaging*, pages 1–11, 2017.

[32] Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. Detect-and-track: Efficient pose estimation in videos. In *International Conference on Computer Vision and Pattern Recognition*, pages 350–359. IEEE, 2018.

[33] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning*, pages 1462–1471. PMLR, 2015.

[34] Banglei Guan, Ji Zhao, Daniel Barath, and Friedrich Fraundorfer. Minimal cases for computing the generalized relative pose using affine correspondences. In *International Conference on Computer Vision*, pages 6068–6077, 2021.

[35] Richard I Hartley. Self-calibration from multiple views with a rotating camera. In *European Conference on Computer Vision*, pages 471–478. Springer, 1994.

[36] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J Black. Resolving 3d human pose ambiguities with 3d scene constraints. In *International Conference on Computer Vision*, pages 2282–2292. IEEE, 2019.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *International Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE, 2016.

[38] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In *Conference on Computer Vision and Pattern Recognition*, pages 7779–7788, 2020.

[39] Markus Hofbauer, Christopher B Kuhn, Jiaming Meng, Goran Petrovic, and Eckehard Steinbach. Multi-view region of interest prediction for autonomous driving using semi-supervised labeling. In *International Conference on Intelligent Transportation Systems*, pages 1–6. IEEE, 2020.

[40] Fuyang Huang, Ailing Zeng, Minhao Liu, Qiuxia Lai, and Qiang Xu. Deepfuse: An imu-aware network for real-time 3d human pose estimation from multi-view image. In *Winter Conference on Applications of Computer Vision*, pages 429–438. IEEE, 2020.

[41] Lin Huang, Jianchao Tan, Ji Liu, and Junsong Yuan. Hand-transformer: Non-autoregressive structured modeling for 3d hand pose estimation. In *European Conference on Computer Vision*, pages 17–33. Springer, 2020.

[42] Lin Huang, Jianchao Tan, Jingjing Meng, Ji Liu, and Junsong Yuan. Hot-net: Non-autoregressive transformer for 3d hand-object pose estimation. In *International Conference on Multimedia*, page 3136–3145. Association for Computing Machinery, 2020.

[43] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*, pages 34–50. Springer, 2016.

[44] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *Transactions on Pattern Analysis and Machine Intelligence*, pages 1325–1339, 2014.

[45] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *Conference on Computer Vision and Pattern Recognition*, pages 7718–7727, 2019.

[46] Ganesh Iyer, R Karnik Ram, J Krishna Murthy, and K Madhava Krishna. Calibnet: Geometrically supervised extrinsic calibration using 3d spatial transformer networks. In *International Conference on Intelligent Robots and Systems*, pages 1110–1117. IEEE, 2018.

[47] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in Neural Information Processing Systems*, 28, 2015.

[48] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *International Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014.

[49] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *British Machine Vision Conference*, pages 12.1–12.11, 2010.

[50] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser S heikh. Panoptic studio: A massively multiview system for social motion capture. In *International Conference on Computer Vision*, pages 3334–3342. IEEE, 2015.

[51] Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical Image Analysis*, pages 61–78, 2017.

[52] Alex Kendall, Roberto Cipolla, et al. Geometric loss functions for camera pose regression with deep learning. In *International Conferences on Computer Vision and Pattern Recognition*, pages 6555–6564, 2017.

[53] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, pages 5580–5590, 2017.

[54] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *International Conference on Computer Vision*, pages 2938–2946. IEEE, 2015.

[55] Cherry Khosla and Baljit Singh Saini. Enhancing performance of deep learning models with different data augmentation techniques: A survey. In *International Conference on Intelligent Engineering and Management*, pages 79–85, 2020.

[56] Hanme Kim, Stefan Leutenegger, and Andrew J Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *European Conference on Computer Vision*, pages 349–364. Springer, 2016.

[57] Brendan F. Klare, Serhat S. Bucak, Anil K. Jain, and Tayfun Akgul. Towards automated caricature recognition. In *International Conference on Biometrics*, pages 139–146. IEEE, 2012.

[58] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *International Conference on Computer Vision and Pattern Recognition*, pages 2041–2050. IEEE, 2018.

[59] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *International Conference on Computer Vision and Pattern Recognition*, pages 1954–1963. IEEE, 2021.

[60] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[61] Yimin Lin, Zhaoxiang Liu, Jianfeng Huang, Chaopeng Wang, Guoguang Du, Jinqiang Bai, and Shiguo Lian. Deep global-relative networks for end-to-end 6-dof visual localization and odometry. In *Pacific Rim International Conference on Artificial Intelligence*, pages 454–467. Springer, 2019.

[62] R. Liu, H. Zhang, M. Liu, X. Xia, and T. Hu. Stereo cameras self-calibration based on sift. In *International Conference on Measuring Technology and Mechatronics Automation*, pages 352–355, April 2009.

[63] Xiaoxia Luo and Feibiao Li. Stacked hourglass networks based on polarized self-attention for human pose estimation. In *Second IYSF Academic Symposium on Artificial Intelligence and Computer Engineering*, pages 543 – 548. International Society for Optics and Photonics, SPIE, 2021.

[64] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, and Zhibin Wang. Tf-pose: Direct human pose estimation with transformers. *arXiv preprint arXiv:2103.15320*, pages 1–15, 2021.

[65] R. B. Mapari and G. Kharat. Real time human pose recognition using leap motion sensor. In *International Conference on Research in Computational Intelligence and Communication Networks*, pages 323–328, 2015.

[66] The MathWorks. *Image Processing and Computer Vision Toolbox*. Natick, Massachusetts, United State, 2022. https://www.mathworks.com/help/overview/.

[67] Stephen J. Maybank and Olivier D. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, pages 123–151, 1992.

[68] Colin J McCarthy and Raul N Uppot. Advances in virtual and augmented reality—exploring the role in health-care education. *Journal of Radiology Nursing*, 38(2):104–105, 2019.

[69] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal V. Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. *2017 International Conference on 3D Vision (3DV)*, pages 506–516, 2016.

[70] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Relative camera pose estimation using convolutional neural networks. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 675–687. Springer, 2017.

[71] Márcio Mendonça, Ivan N Da Silva, and José EC Castanho. Camera calibration using neural networks. *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 61–64, 2002.

[72] Eric N Mortensen, Hongli Deng, and Linda Shapiro. A sift descriptor with global context. In *International Conference on Computer Vision and Pattern Recognition*, pages 184–190. IEEE, 2005.

[73] Pierre Moulon, Pascal Monasse, Renaud Marlet, and Others. Openmvg. an open multiple view geometry library. https://github.com/openMVG/openMVG, 2016.

[74] Etienne Mouragnon, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser, and Patrick Sayd. Monocular vision based slam for mobile robots. In *International Conference on Pattern Recognition*, volume 3, pages 1027–1031. IEEE, 2006.

[75] Jorge Nelson Neves, Pedro Gonçalves, Joaquim Muchaxo, and João P Silva. A virtual gis room: interfacing spatial information in virtual environments. In *Spatial Multimedia and Virtual Reality*, pages 149–158. CRC Press, 2021.

[76] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[77] Juan Carlos Núñez, Raúl Cabido, José F Vélez, Antonio S Montemayor, and Juan José Pantrigo. Multiview 3d human pose estimation using improved least-squares and lstm networks. *Neurocomputing*, pages 335–343, 2019.

[78] Andrew O'Riordan, Thomas Newe, Gerard Dooly, and Daniel Toal. Stereo vision sensing: Review of existing systems. In *International Conference on Sensing Technology*, pages 178–184, 2018.

[79] PhaseSpace. 3d character creation. https://www.phasespace.com/applications/3dcharactercreation/, 2022. Accessed: (03-02-2022).

[80] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *International Conference on Computer Vision and Pattern Recognition*, pages 588–595. IEEE, 2013.

[81] Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. Cross view fusion for 3d human pose estimation. In *International Conference on Computer Vision*, pages 4342–4351. IEEE, 2019.

[82] Weichao Qiu and Alan Yuille. Unrealcv: Connecting computer vision to unreal engine. In *European Conference on Computer Vision*, pages 909–916. Springer, 2016.

[83] Edoardo Remelli, Shangchen Han, Sina Honari, Pascal Fua, and Robert Wang. Lightweight multi-view 3d pose estimation through camera-disentangled representation. In *International Conference on Computer Vision and Pattern Recognition*, pages 6040–6049. IEEE, 2020.

[84] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *European Conference on Computer Vision*, pages 750–767. Springer International Publishing, 2018.

[85] Helge Rhodin, Jörg Spörri, Isinsu Katircioglu, Victor Constantin, Frédéric Meyer, Erich Müller, Mathieu Salzmann, and Pascal Fua. Learning monocular 3d human pose estimation from multi-view images. In *International Conference on Computer Vision and Pattern Recognition*, pages 8437–8446. IEEE, 2018.

[86] Rafael E Rivadeneira, Patricia L Suárez, Angel D Sappa, and Boris X Vintimilla. Thermal image superresolution through deep convolutional neural network. In *International Conference on Image Analysis and Recognition*, pages 417–426. Springer, 2019.

[87] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *International Conference on Computer Vision*, pages 2564–2571. IEEE, 2011.

[88] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal Computer Vision*, page 211–252, 2015.

[89] Benjamin Sapp, Chris Jordan, and Ben Taskar. Adaptive pose priors for pictorial structures. In *International Conference on Computer Vision and Pattern Recognition*, pages 422–429. IEEE, 2010.

[90] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *International Conference on Computer Vision and Pattern Recognition*, pages 3247–3257. IEEE, 2021.

[91] Hamid Sarmadi, Rafael Muñoz-Salinas, MA Berbís, and RJIA Medina-Carnicer. Simultaneous multi-view camera pose estimation and object tracking with squared planar markers. *IEEE Access*, pages 22927–22940, 2019.

[92] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions. In *International Conference on Computer Vision and Pattern Recognition*, pages 8601–8610. IEEE, 2018.

[93] Shreya Saxena and Jakob Verbeek. Heterogeneous face recognition with cnns. In *European Conference Computer Vision*, pages 483–491, 2016.

[94] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *International Conference on Computer Vision and Pattern Recognition*, pages 4104–4113. IEEE, 2016.

[95] EV Shalnov and AS Konushin. Convolutional neural network for camera pose estimation from object detections. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, pages 1–6, 2017.

[96] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *International Conference on Computer Vision and Pattern Recognition*, pages 2930–2937. IEEE, 2013.

[97] Hui Shuai, Lele Wu, and Qingshan Liu. Adaptively multi-view and temporal fusing transformer for 3d human pose estimation. *ArXiv*, abs/2110.05092, 2021.

[98] Rafał Sieczka and Maciej Pańczyk. Blender as a tool for generating synthetic data. *Journal of Computer Sciences Institute*, 16:227–232, 2020.

[99] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Computing Research Repository*, abs/1409.1556, 2014.

[100] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations*, 2015.

[101] Jun Sun, Mantao Wang, Xin Zhao, and Dejun Zhang. Multi-view pose generator based on deep learning for monocular 3d human pose estimation. *Symmetry*, 12(7), 2020.

[102] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *International Conference on Computer Vision and Pattern Recognition*, pages 5686–5696. IEEE, 2019.

[103] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *International Conference on Computer Vision*, pages 2621–2630. IEEE, 2017.

[104] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning*, pages 1139–1147. Machine Learning Research, Jun 2013.

[105] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *International Conference on Computer Vision and Pattern Recognition*, pages 1–9. IEEE, 2015.

[106] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, D. Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *International Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[107] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. Data augmentation using random image cropping and patching for deep cnns. *Circuits and Systems for Video Technology*, 30:2917–2931, 2018.

[108] Cong Tang, Yongshun Ling, Xing Yang, Wei Jin, and Chao Zheng. Multi-view object detection based on deep learning. *Applied Sciences*, page 1423, 2018.

[109] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. In *International Conference on Learning Representations*, pages 1–18, 2020.

[110] Chunwei Tian, Yong Xu, Lunke Fei, Junqian Wang, Jie Wen, and Nan Luo. Enhanced cnn for image denoising. *Transactions on Intelligence Technology*, 4(1):17–23, 2019.

[111] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *International Conference on Computer Vision and Pattern Recognition*, pages 648–656. IEEE, 2015.

[112] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *International Conference on Computer Vision and Pattern Recognition*, pages 1653–1660. IEEE, 2014.

[113] Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *British Machine Vision Conference*, pages 1–13, 2017.

[114] Ben Usman, Andrea Tagliasacchi, Kate Saenko, and Avneesh Sud. Metapose: Fast 3d pose from multiple views without 3d supervision. In *International Conference on Computer Vision and Pattern Recognition*, pages 6759–6770. IEEE, June 2022.

[115] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[116] Ziniu Wan, Zhengjia Li, Maoqing Tian, Jianbo Liu, Shuai Yi, and Hongsheng Li. Encoder-decoder with multi-level attention for 3d human shape and pose estimation. In *International Conference on Computer Vision*, pages 13013–13022. IEEE, 2021.

[117] Bastian Wandt, Marco Rudolph, Petrissa Zell, Helge Rhodin, and Bodo Rosenhahn. Canonpose: Self-supervised monocular 3d human pose estimation in the wild. *International Conference on Computer Vision and Pattern Recognition*, pages 13289–13299, 2021.

[118] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *International Conference on Computer Vision and Pattern Recognition*, pages 5459–5470. IEEE, 2021.

[119] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *International Conference on Computer Vision and Pattern Recognition*, pages 2285–2294. IEEE, 2016.

[120] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, pages 135–153, 2018.

[121] Tao Wang, Jianfeng Zhang, Yujun Cai, Shuicheng Yan, and Jiashi Feng. Direct multi-view multi-person 3d pose estimation. In *Advances in Neural Information Processing Systems*, pages 13153–13164. Curran Associates, Inc., 2021.

[122] Xiangyang Wang, Jiangwei Tong, and Rui Wang. Attention refined network for human pose estimation. *Neural Processing Letters*, page 2853–2872, 2021.

[123] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *International Conference on Computer Vision and Pattern Recognition*, pages 8737–8746. IEEE, 2021.

[124] Tom Wehrbein, Marco Rudolph, Bodo Rosenhahn, and Bastian Wandt. Probabilistic monocular 3d human pose estimation with normalizing flows. In *International Conference on Computer Vision*, pages 11199–11208. IEEE, October 2021.

[125] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *International Conference on Computer Vision and Pattern Recognition*, pages 4724–4732. IEEE, 2016.

[126] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In-So Kweon. Cbam: Convolutional block attention module. In *European Conference on Computer Vision*, 2018.

[127] Minghu Wu, Hanhui Yue, Juan Wang, Yongxi Huang, Min Liu, Yuhan Jiang, Cong Ke, and Cheng Zeng. Object detection based on rgc mask r-cnn. *Image Processing*, 14(8):1502–1508, 2020.

[128] Hailun Xia and Tianyang Zhang. Self-attention network for human pose estimation. *Applied Sciences*, 2021.

[129] L. Xia, C. Chen, and J. K. Aggarwal. Human detection using depth information by kinect. In *International Conference on Computer Vision and Pattern Recognition*, pages 15–22, 2011.

[130] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision (ECCV)*, pages 466–481. Springer, 2018.

[131] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *International Conference on Computer Vision and Pattern Recognition*, pages 2690–2698, 2019.

[132] Jingming Xie. Research on key technologies base unity3d game engine. In *International Conference on Computer Science & Education*, pages 695–699. IEEE, 2012.

[133] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057. PMLR, 2015.

[134] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Keypoint localization via transformer. *International Conference on Computer Vision*, pages 11782–11792, 2021.

[135] Zhenbo Yu, Bingbing Ni, Jingwei Xu, Junjie Wang, Chenglong Zhao, and Wenjun Zhang. Towards alleviating the modeling ambiguity of unsupervised monocular 3d human pose estimation. In *International Conference on Computer Vision*, pages 8651–8660. IEEE, October 2021.

[136] Zijun Zhang. Improved adam optimizer for deep neural networks. In *International Symposium on Quality of Service*, pages 1–2, 2018.

[137] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *International Conference on Computer Vision*, pages 11636–11645. IEEE, 2021.

[138] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Conference on Artificial Intelligence*, pages 13001–13008, 2017.

[139] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. *Advances in Neural Information Processing Systems*, 27:487–495, 2014.

[140] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, pages 1–16, 2021.