PREMIERE
Performing arts in a new era

3D analysis understanding and reconstruction v1

December 2023 - M15

Document identifier: PRMR-D4.1-3D analysis understanding and reconstruction v1
Version: 1.0
Author: UJM

Funded by
the European Union

Dissemination status: PU

Grant Agreement n°	101061303
Project acronym	PREMIERE
Project title	Performing arts in a new era: AI and XR tools for better understanding, preservation, enjoyment and accessibility
Funding Scheme	HORIZON-CL2-2021-HERITAGE-01 (HORIZON Research and Innovation Actions)
Project Duration	01/10/2022 – 30/09/2025 (36 months)
Coordinator	Athena Research Center (ARC)
Associated Beneficiaries	• Stichting Amsterdamse Hogeschool voor de Kunsten (AHK) • Forum Danca - Associacao Cultural (FDA) • Tempesta Media SL (TMP) • Cyens - Centre of Excellence (CNS) • Kallitechniki Etaireia Argo (ARG) • Medidata.Net - Sistemas de Informacao para Autarquias SA (MED) • Fitei Festival Internacional Teatro Expresao Iberica Crl (FIT) • Instituto Stocos (STO) • Universite Jean Monnet Saint-Etienne (UJM) • Associacao dos Amigos do Coliseu Doporto (COL) • Stichting International Choreographic Arts Centre (ICK)

Funded by the European Union

Project no. 101061303
PREMIERE

Performing arts in a new era: AI and XR tools for better understanding, preservation, enjoyment and accessibility

HORIZON-CL2-2021-HERITAGE-01

Start date of project: 01/10/2022

Duration: 36 months

History Chart				
Issue	**Date**	**Changed page(s)**	**Cause of change**	**Implemented by**
0.1	22/11/2023	-	Draft	UJM
0.2	01/12/2023	-	1st version	UJM
0.3	08/01/2024	ALL	Revised (final) version	UJM
1.0	23/01/2024	ALL	Camera ready version	UJM

Validation			
No.	**Action**	**Beneficiary**	**Date**
1	Prepared	Aggelos Gkiokas (ARC) Panagiotis Charalampous (CNS)	05/01/2024
2	Approved	Aggelos Gkiokas (ARC) Panagiotis Charalampous (CNS)	09/01/2024
3	Released	Aggelos Gkiokas (ARC) Panagiotis Charalampous (CNS)	23/01/2024

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them.

Table of Contents

Executive Summary

This deliverable provides an overview of Tasks 4.1 and 4.2 within the PREMIERE project. **Task T4.1 - 3D scene Analysis and Understanding** is structured on three sub-tasks. The aim of **sub-task T4.1a** is to create a dataset of live performances, representatives of complex dance or theatre events (with multi-person), acquired with several cameras under different points of views. The annotation (labelling) of these audio-visual contents will enable us to test, compare and retrain Convolutional Neural Networks (CNN) and other Deep Learning methods. The aim of **sub-task T4.1b** is to build a 3D model of moving people in these audio-visual contents from the different views available and to evaluate the potential of the methods investigated and implemented. The most promising 3D scene analysis and understanding methods (such as multi-people pose and motion estimation methods, segmentation and tracking methods, action parsing methods, etc.) have been tested, compared and evaluated. Experimental results obtained from single-view methods (and archives) or multi-views methods have been compared and analysed. The aim of **sub-task T4.1c** is to build a 3D model of static elements present in audio-visual archives. The most efficient 3D scene analysis and understanding methods (such as object detection methods, segmentation and tracking methods, etc.) have been tested, compared and evaluated. In the next six months (i.e. month M21 of the project), these 3D representations of static elements will be integrated as additional data for the algorithms used in the analysis, understanding and 3D reconstruction of audio-visual contents.

The objective of Task 4.1 is to define a set of generic methods/tools that could be generalised/transferred to other audio-visual contents (or study cases) than the ones investigated in the duration of the PREMIERE project, depending of the complexity of their audio-video content, and to evaluate the efficiency and performance of the methods investigated and implemented in order to define a list of rules (linked to the content of the events, to the costume elements, to the lighting conditions, to the quality of the data, etc.) regarding the set of functionalities that could be extended to other events, or need some adaptations, or cannot be used.

T4.2 - 3D pose trajectories estimation in complex scenes is structured on four sub-tasks. We have investigated the potential of individual object (such as human body, hands) pose and motion estimation methods (**sub-task T4.2a**), human body pose trajectories estimation methods (**sub-task T4.2b**), tracking methods (based on 3D keypoint detection, kinematic skeleton, bounding box, semantic segmentation, 3D template model) (**sub-task T4.2c**), action parsing methods, etc., and have investigated/developed solutions to handle inter-object occlusions, abrupt motion changes, appearance changes due to different lighting conditions between points of views, low contrast with the background, missing detections due to non-rigid deformation of clothing, truncations of persons, or interactions between people, etc. (sub-task T4.2d). In the next six months, the methods that will be investigated will integrate as a constraint the 3D representation of the static set elements obtained in T4.1c and the need for the human avatars modelling (task T4.3).

As for Task 4.1, the objective of Task 4.2 is to define a set of generic methods/tools that could be generalised/transferred to other audio-visual contents (or study cases) than the first videos investigated in the duration of the PREMIERE project. The objective is also to evaluate the efficiency and performance of the methods investigated and implemented (in function of the complexity of the video content, of the costume elements, of the lighting conditions, of the quality of the data, etc.), in order to identify the set of functionalities that can be extended to any video content, or which need some adaptations depending of the video content, or which will not work properly for some video contents.

As Task 4.1 and 4.2 are intrinsically inter-connected their description is mixed in the following sections.

Our ability to use in an optimal way, to parameterize and to optimize, deep learning networks and models relies heavily on a diverse set of cutting-edge architectures, algorithms and tools encompassing a wide range of video processing tasks, such as 2D pose estimation, 3D pose estimation, 3D tracking, 3D trajectories estimation, etc. Meanwhile the efficiency of the architectures, algorithms and tools, selected have been demonstrated in the State-Of-the-Art, their robustness to occlusion, motion blur, frame rate, noise, etc. has not been fully investigated, especially in the context of contemporary dance. Our objective was therefore to propose a framework that leverage the limits of existing approaches. Our strategy is to propose to end-users a framework that automatically process a video file, and to give to end-users the possibility to use a toolbox to improve the processing tasks / to refine the results obtained. To achieve this goal, we will delve into cutting-edge deep learning architectures and conventional video processing algorithms. This will require also to setup efficient accuracy metrics.

In summary, our pragmatical approach combines diverse cutting-edge deep learning architectures, few conventional video processing algorithms, and an innovative video processing framework, to advance our analysis and modelling of 3D scenes and of dancers' motion. This research has the potential to find applications in fields such as animation, artistic creation, dance schools, and beyond.

Acronyms and abbreviations

Abbreviation	Description
AI	Artificial Intelligence
CNN	Convolutional Neural Network
JSON	JavaScript Object Notation
LMA	Laban Movement Analysis
ViT	Vision Transformer

1. Introduction

This deliverable focuses on Task T4.1 - 3D scene Analysis and Understanding and T4.2 - 3D pose trajectories estimation in complex scenes. As Tasks 4.1 and 4.2 are intrinsically inter-connected their description is mixed in the following sections. The main output of this deliverable is a video processing pipeline consisting of 13 video processing modules structured as follow (see Figure 1). Section 2 details the objectives defined, the architectures and models selected, the tools and algorithms implemented, the input and outputs, for all these modules. Section 3 concludes this deliverable.

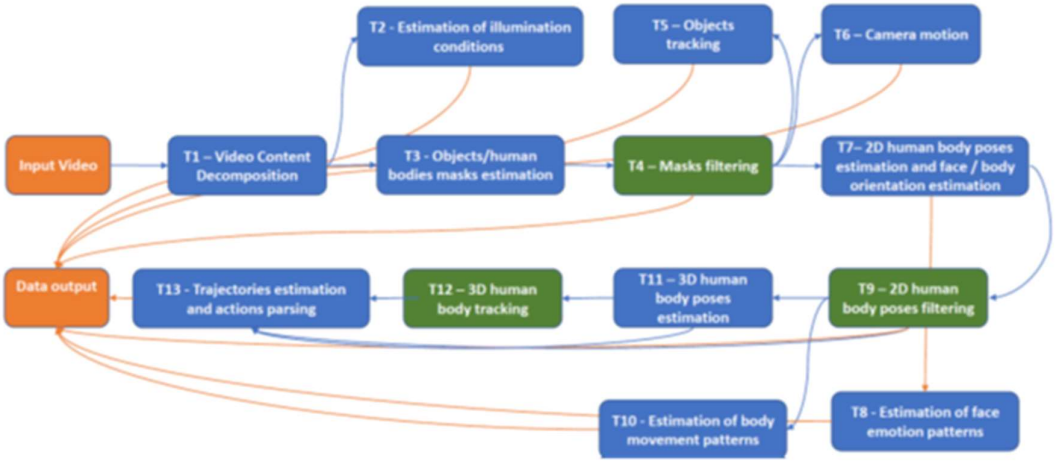


Figure 1 : The modular video processing pipeline

Funded by the European Union

2. 3D Scene analysis and understanding

The first goal of this deliverable is to provide a set of architectures and models, and a set of tools and algorithms, designed for: (a) analysing a video content, (b) estimating 2D human body pose, (c) estimating 3D human body pose and shape, (d) tracking 3D human body, (e) estimating human body trajectories and parsing actions. Section 2.2 focuses on **Tasks 1 to 6** of the video processing pipeline illustrated in Figure 1. Section 2.3 focuses on **Tasks 7 to 10.** Section 2.4 focuses on **Task 11**. Section 2.5 focuses on **Task 12**. Section 2.6 focuses on **Task 13.**

The second goal of this deliverable is to provide solutions to test, compare and evaluate the efficiency and robustness of existing tools. Section 2.1 focuses on the creation of an annotated life performance dataset built for this purpose.

2.1. Creation of an annotated life performance dataset (Task T4.1a of the WP4)

The objective is to create a life performance dataset of video contents with specific features (described after in Table 1). To the best of our knowledge, the AIST++ Dance Motion Dataset [1] is the only existing dance motion dataset that meets most of the expected criteria (summarized in Table 1) but it doesn't satisfy all of them, that is why we decided to enrich this dataset with another dataset. The AIST++ Dance Motion Dataset contains 1,408 human dance motion sequences on 10 dance genres with hundreds of choreographies, acquired from 9 views. It provides camera intrinsic and extrinsic parameters computed from a 3D calibration process based on keypoints. Motion durations vary from 7.4 sec. to 48.0 sec. In total, this dataset contains 10,108,015 frames of 3D keypoints with corresponding images. 3D human body modelling is done in SMPL format. The code for 3D reconstruction 3D calibration, etc. is available online[1] . For this task, we have acquired new short clips of dancers (PREMIERE Dance Motion Dataset) under specific and challenging conditions (see Table 1) not available in the AIST++ dataset.

Video features	AIST++ Dance Motion Dataset	PREMIERE Dance Motion Dataset
Multi-views synchronised images	9 cameras	4 cameras (see Figure 2).
Occlusions between dancers or auto-occlusion	1, 2 or 10 dancers (depending on the sequence) [2]. Few occlusions or auto-occlusion.	1 or 2 dancers (depending on the sequence) with complex occlusion patterns. Specific occlusion features (e.g. see Figure 3).
Unconventional pose, velocity and acceleration	Floor dance genre: Break, Pop, Lock and Waack, Middle Hip-hop, LA-style Hip-hop, House, Krump, Street Jazz and Ballet Jazz (basic and advanced)	Unconventional pose (e.g. see Figure 4). Motion blur. Contemporary dance/choreography
Heterogeneous lighting conditions	Constant lighting conditions (white source) with low shading	Strong variations of lighting conditions over time and over areas (e.g. see Figure 5). Strong variations of shading.

[1] https://github.com/google/aistplusplus_api

Lack of discriminative features between dancers (except faces)	Few dancers wear dark clothes	Dancers wear dark clothes
Specific features	In very few videos, dancers express various face emotions. Image size 1920x1080 pixels, 60 FPS, encoding MP4/RAW	In few videos, dancers express various face emotions (e.g. see Figure 17). Image size 5312x2988 pixels, 60 FPS, encoding MP4

Table 1: Specific features that make pose estimation challenging.

Figure 2: Images of the same video sequence from different viewpoint. (a) from front camera (the best view to estimate the orientation of body shapes and faces emotion), (b) same timestamp from side view (strong occlusion between dancers), (c) same timestamp from the back. Images from the PREMIERE Dance Motion Dataset.

Figure 3 : Images the PREMIERE Dance Motion Dataset. (a) one of the dancers occludes the lower part of the body of the second dancer, (b) complex occlusion pattern, (c) another challenging occlusion pattern.

Figure 4: Images from the PREMIERE Dance Motion Dataset. (a) The feet of the dancers are not connected to the ground, (b) motion blur for one of the hands, (c) complex pose of the dancers.

Figure 5: Images from another video sequence taken using with two asynchronous color spots. (a) dancers are in the shadow area, (b) color shifts induced by changes of lighting color over time, (c) complex shading effects induced by changes of lighting color over time. Images from the PREMIERE Dance Motion Dataset.

This dataset will enable us to evaluate the efficiency/robustness of human body pose estimation methods against specific features, such as the ones reported in the Table 1. As example see results shown in Figure 6.

Figure 6: Examples of 3D reconstruction of human bodies (obtained from HMR 2.0) for complex scenes. Images from the PREMIERE Dance Motion Dataset.

2.2. 2D video content analysis (Task 4.1c of the WP4)

2.2.1. Video content decomposition

The **first task** to analyse a video content (Task T1 of the image processing pipeline illustrated in Figure 1) is to automatically cut the video into individual scenes (i.e. to detect scene content changes). It exists in the State-Of-the-Art efficient methods to detect shot changes in videos, such as PySceneDetect1. As an example of results, see Figure 7.

- Input: video file with all scenes (format H264 video or JSON)
- Output: video file split in individual scenes (format H264 video or JSON)

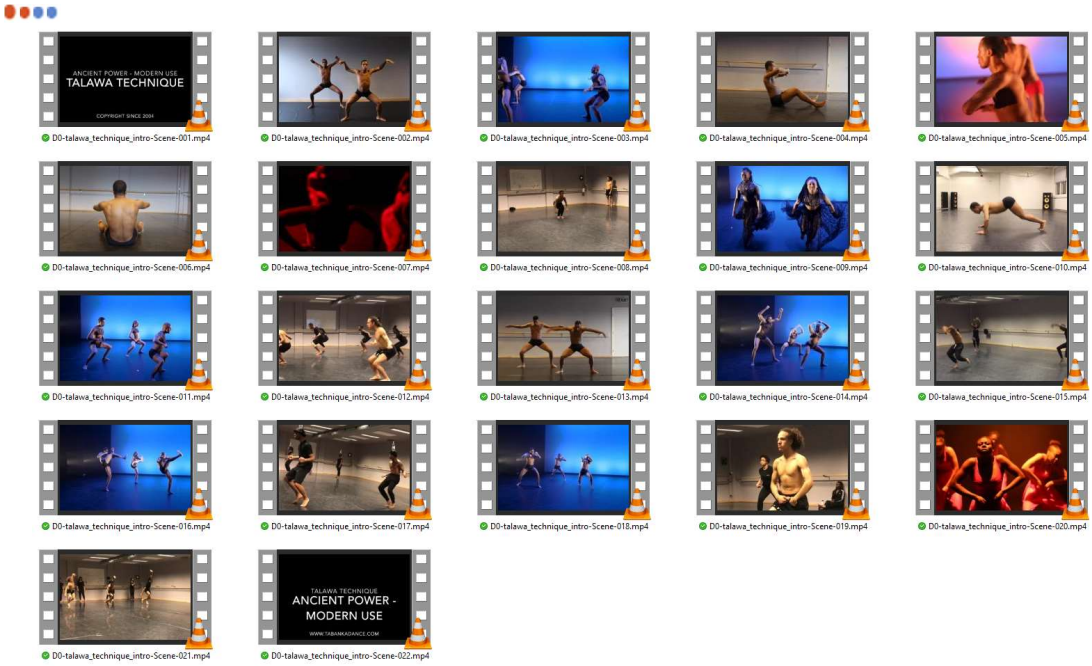


Figure 7: Individual sequences detected in the Talawa video.

2.2.2. Estimation of illumination conditions

The second task to analyse a video content (Task T2 of the image processing pipeline illustrated in Figure 1) is to estimate the illumination conditions of each individual sequence. When the lighting conditions are too dark (as in sequence 7 of Talawa video, see Figure 7), when there is too much shadow (as in the first image of Figure 5), when the lighting field is too colourful (as in sequence 20 of Talawa video, see Figure 7) or diverse (as in the last image of Figure 5), object detection is very challenging. Most of objects detection methods are robust against lighting variations up to a certain degree. The objective is to estimate if the

lighting conditions are good enough to perform objects detection (i.e. to check if the color contrast between each object of interest and its background is enough high). The estimation of lighting conditions can be also useful to perform color correction of the lighting field (to better balance the impact of different light sources, to improve the efficiency of objects detection, to modify the color rendering of video contents, etc. (as example see [3])). Finally, the use cases of our project require to extract the lighting conditions of a performance to classify performances with respect to their lighting conditions, compare these conditions between performances, to re-render the scene under different lightings, etc. State-Of-the-Art methods exist to estimate illumination conditions of scenes lighted by a single light source, such as IndoorLightEditing2. We are going to test this method and related works to select the best solution for our context.

- Input: video file with all scenes (format H264 video or JSON)
- Output: video file split in individual scenes (format H264 video or JSON) with for each frame a set of image maps (shadow map, spatial illumination map) and a set of information about lighting sources (position in the space, chromaticity of the illuminants, etc.)

2.2.3. Image segmentation

The third task (Task T3 of the image processing pipeline illustrated in Figure 1) consists in automatically segmenting all people and all objects in each video frame (as example see Figure 8). It exists in the State-Of-the-Art efficient methods to detect objects and people in video frames, but errors can occur in some study cases (eg. when objects/people are in the dark, when the color of the objects/people is insufficiently contrasted with the background, when unusual objects must be detected, etc. (see Figures 8 and 9). The "Segment Everything Everywhere All at Once" (SEEM) method3 provides very good segmentation results for people, but if the input images exceed a certain size their resolution must be reduced. SEEM outperforms other segmentation methods in position accuracy, occlusion accuracy, and temporal coherence across different datasets [4]. SEEM allows full-length motion estimation for every pixel in a video. It tackles the challenges of motion estimation by ensuring global consistency, tracking through occlusions, and handling in-the-wild videos with any combination of camera and scene motion4. SEEM struggles with rapid and highly non-rigid motion and thin structures. In challenging scenarios SEEM may not provide enough reliable correspondences as it relies on pairwise correspondences. SEEM is computationally expensive. The "Grounded Segment Anything" method5 combining the "Segment Anything in High Quality" (SAM HQ) technic [5] and the "Grounding DINO" technic6 provides very good objects' detection results. Several other efficient methods exist in the State-Of-the-Art (see [6] as an example), the heavier are in general more accurate than the lighter.

- Input: individual scenes (format H264 video or JSON) (output from Task 1)
- Output: individual scenes (format H264 video or JSON, and Python binary format - pickle) with for each frame a set of masks or a set 2D bounding boxes (one for each object detected).

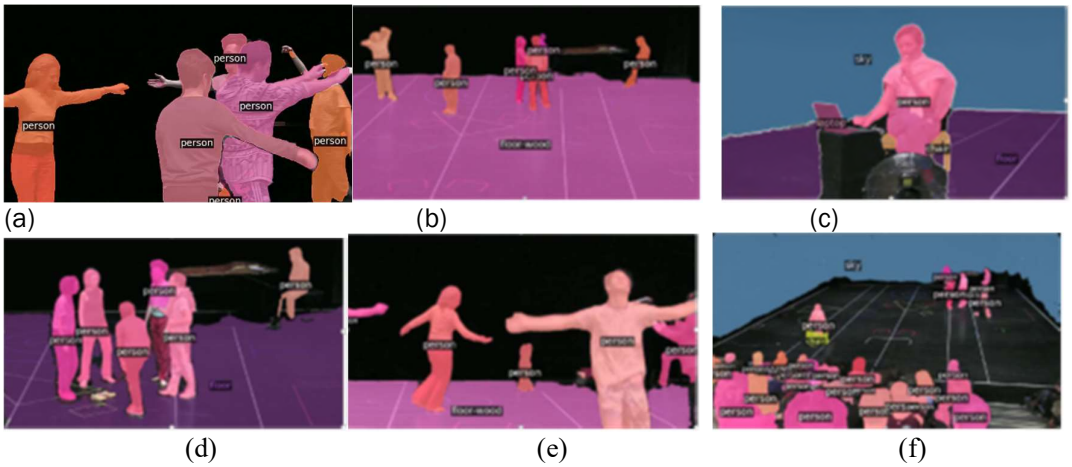


Figure 8: Examples of individual frames automatically segmented. Most of people are accurately detected, meanwhile most of objects are not detected (e.g. the piano in frame (b), the fan in frame (c), the shoes on the ground in frame (d)). Nevertheless, in frame (d) one of the people is partially detected; in frame (e) one of the feet of the piano is perceived as a person; in frame (f), few errors of detection can be noticed in the public area. Note that in the background the intensity of the lighting field is very low which makes the object detection more challenging. With each object is associated a label (e.g. a person, a floor, etc.) and a mask. Output images are here coded in Python binary format (i.e. as greyscale image and a Look-Up-Table).

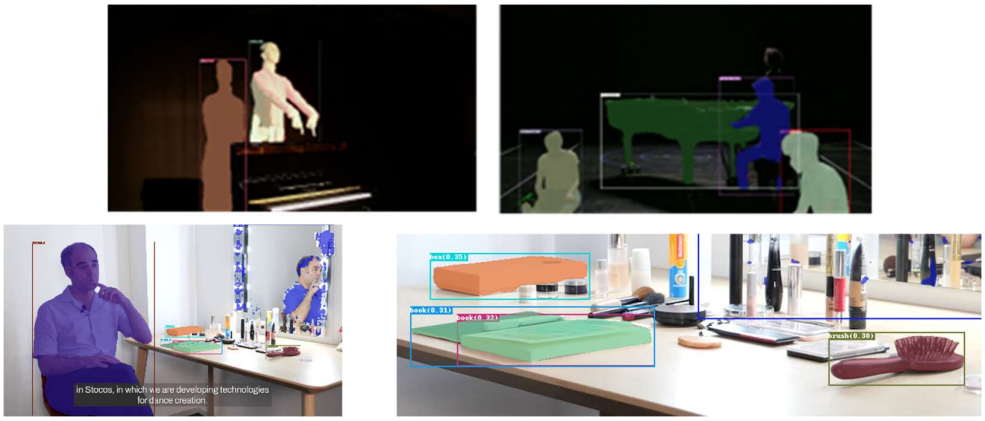


Figure 9: At each segment of a video frame, we can attach an instance. (ie. a class, eg. a piano, a brush, a textual area) and a bounding box. Note that the object detection task is are more challenging for small objects or some objects' classes (see bottom images).

2.2.4. Segmentation refinement

The fourth task to analyse a video content (Task T4 of the image processing pipeline illustrated in Figure 1) is to filter the sets of masks computed using temporal information (previous and subsequent masks) when necessary, as example see Figure 10. The objective is to maintain accurate tracking across long sequences and to remove incoherent masks over time. Several efficient methods exist in the State-Of-the-Art, such as AuxAdapt [7, 8]. Depending on the object features (some objects are easier to detect than other), on the video content (some objects are rapidly moving, others are in a fixed position), and on the temporal consistency of the video content (some frames can be sometime noisy, some objects can be partially occluded), some methods can perform better than others.

- Input: individual scenes (format H264 video or JSON, and Python binary format - pickle) with for each frame a set of masks (output from Task 3)

- Output: individual scenes (format H264 video or JSON, and Python binary format - pickle) with for each frame a set of masks.

Figure 10: Examples of incoherent masks over time (few ms). (a) On the top video sequence: the chair is missing on the last image; the shirt is out of the human body mask on the second image. (b) On the bottom video sequence: the fan is detected as a chair on the second image.

2.2.5. Objects tracking

The fifth task (Tasks T5 and T6 of the image processing pipeline illustrated in Figure 1) is to refine the tracking of moving objects in a video sequence from the optical flow, when necessary. The tracking of moving objects can be inaccurate when the speed of the motion exceeds the frame rate of the camera (in this case we have motion blur)7, and when occlusions or camera motion affect significantly the results of the objects tracking (see Figure 11 as an example). Several efficient methods exist in the State-Of-the-Art, such as CoTracker8.

- Input: individual scenes (format H264 video or JSON) (output from Task 4)
- Output: individual scenes (format H264 video or JSON) with for each frame a map of track points.

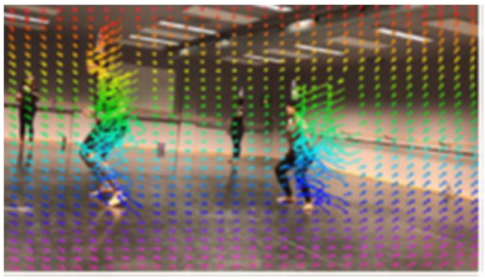


Figure 11: video sequence 17 of the Talawa video.The optical flow map highlights the camera motion over time. The motion estimation of the two dancers at the forefront is impacted by the motion of the camera.

2.3. 2D human pose estimation analysis

2.3.1. Human body pose estimation

The first task to estimate human body pose in a video content (Task T7 of the image processing pipeline illustrated in Figure 1) is to automatically estimate human body pose from human keypoints9 (body + face + hands and feet) in 2D. It exists in the State-Of-the-Art

efficient methods to estimate human body pose in 2D, such as ViTPose+10 or DWPose11. ViTPose++ provides very good results and performs better than most of State-Of-the-Art methods, such as AlphaPose12 or MMpose13. The heavier pose estimation methods are in general more accurate than the lighter. Depending on the human features (some keypoints/segments are easier to detect than others, see Figure 12) or on the video content (some keypoints/segments are harder to detect than others, see Figures 13 and 14), some methods can perform better than others. Several keypoint representations can be used to model a human body, the number of joints and of keypoints for each instance varies from one model to another one (see [10]). Some architectures, such as VITPose++ can deal with several keypoint representations. Several metrics can be used to evaluate the accuracy of a pose estimation method, such as the Percentage of Detected Joints (PDJ)14 for the whole body, the Object Keypoint Similarity (OKS)15 for each body part, the Percentage of Correct Parts (or Correctly Estimated Body Parts)16 – PCP for each body part of a set of images, Percentage of Correct Key-points - PCK17, the Mean Per Joint Position Error - MPJPE18, and the Multi-Instance Mean Squared Error (see [9]). These metrics require to know the true position of keypoints (i.e. to have access to geometrically calibrated images). It has been demonstrated that occlusion and truncation have an impact on the accuracy. Torso viewpoint, part length (foreshortening) and activity have also an impact on the accuracy (see [11]). The limits of UpToDate pose estimation methods are demonstrated by detection errors that occur when these later are applied to the most complex videos of the PREMIERE dataset.

Figure 12: Examples of wrong pose estimation due to the dressing of the dancers and low light conditions. ***(video sequence N°9 of Talawa video processed with Alphapose).***

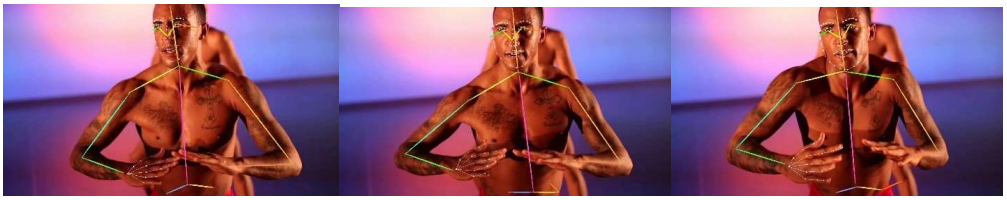


Figure 13: Examples of inaccurate pose estimation of the face and hands of the dancer even if the dancer is in front of the camera. ***In this sequence, only the upper body of dancers is available. Note that the spatial resolution is here sufficient to analyze face emotion from face keypoints (video sequence N°5 of Talawa video processed with Alphapose).***

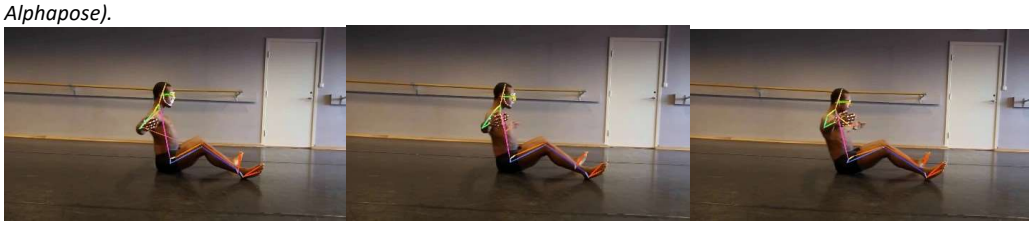


Figure 14: Examples of accurate pose estimation of the right hand and of the two legs of the dancer even if the dancer is sitting perpendicularly to the view of the camera. ***(part of his body shape is therefore occluded) (video sequence N°4 of Talawa video processed with Alphapose)***

One way to reduce detection errors is to play with the confidence score of each body part (e.g. to remove hands and face keypoints when these data have high uncertainties, or irrelevant predictions as in Figure 12) or of the whole body (e.g. to remove ghost keypoints induced by human body shadows as in Figure 15).

Figure 15: Example of ghost keypoints detected in the shadow of the standalone dancer. Image from the Talawa video.

- Input: individual scenes (format H264 video or JSON, and Python binary format - pickle) (output from **Task 4**) with for each frame a set of masks (one for each people)
- Output: individual scenes (format H264 video or JSON, and Python binary format - pickle) with for each frame a set of keypoints for each person, and a 2D bounding box (with 2D position).

2.3.2. Human body orientation estimation

The second task is to automatically estimate human body orientation from human body keypoints (from torso keypoints and shoulders keypoints). Body orientation provides useful information to characterize pose configuration and to solve self-occlusion problems or left-right similarity problems. For example, if we know a person is facing to right, the body orientation indicates the occlusion of his or her left body. Similarly, if a person is facing the camera, his or her right shoulder is probably on the left side of the image. The body orientation can be defined by a vector format with 8 elements (see illustration from Figure 16) and a confidence score. It exists in the State-Of-the-Art efficient methods to estimate human body orientation, such as IntePoseNet19 or MEBOW20 (see [12]). In dance, the torso is not necessarily aligned with the face, so it does not make sense to use face keypoints to characterize human body pose.

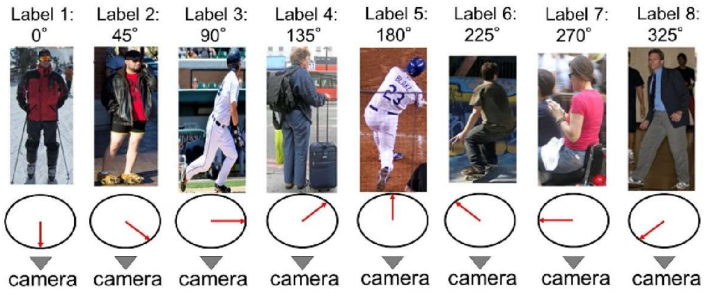


Figure 16/ The orientation of a human body defined by 8 viewing directions of 45° each. Image from [13].

From one view to another one (as example see Figure 2) the accuracy of the human body orientation prediction may change a lot. The best view is in general the one for which the confidence score is the upper (when the people are in front of the camera).

From one view to another one, the accuracy of the pose estimation may change a lot. The best view is in general the one for which the confidence score is the upper (when the people are in front of the camera). It is more challenging to characterize the best view for hands pose

estimation, as the two hands can be oriented in different directions (as illustration see Figure 13).

The best view for face emotion estimation for which the confidence score of face keypoints is the upper is in general the one for which the faces are in front of the camera (as illustration see Figure 13).

- Input: individual scenes (format H264 video or JSON, and Python binary format - pickle) with for each frame a set of keypoints for each body part, and a 2D bounding box (with 2D position). (output from Task 7)
- Output: confidence score for each set of keypoints and an orientation value for torso, face and hands (for each view)

2.3.3. Face emotion recognition

The third task is to automatically recognise face emotion (Task T8 of the image processing pipeline illustrated in Figure 1, connected to Task 3.2 of the WP3). It exists in the State-Of-the-Art efficient methods to recognise face emotion21. The accuracy of the estimation depends on the spatial resolution of the images and of the image size of face areas (see Figure 17). In this context, the main challenge is in the generalization power of the trained model. Indeed, after training a deep network to recognize face emotion, it appears that the results at test time are good for images representing the same people in images acquired under the same conditions. We are working on domain generalization for this task.

- Input: individual scenes (format H264 video or JSON, and Python binary format - pickle) with for each frame a set of keypoints for each face detected and a 2D bounding box (with 2D position). (output from Task 7)
- Output: a probability distribution over the face emotion classes for each bounding box.

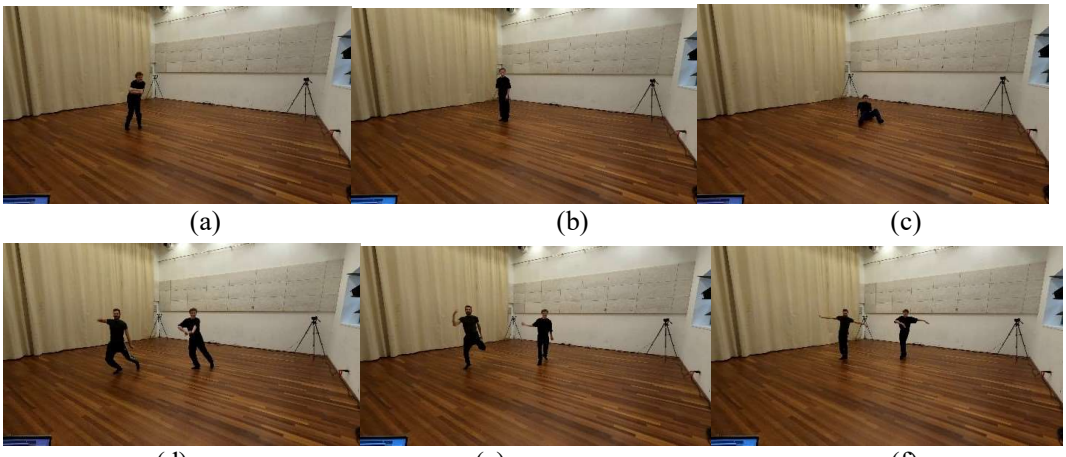


Figure 17: Face emotion patterns. On the top: Images from one video sequence (image size 5312x2988 pixels). (a) dancer smiling (face size 131x161 pixels), (b) dancer laughing (face size 95x127 pixels), (c) neutral emotion (face size 89x109 pixels). Image (a) is the closest to the viewing camera, image (c) the farthest.On the bottom: Images from one video sequence (image size 5312x2988 pixels). (d) dancers laughing (smallest face size 117x150 pixels), (e) dancers smiling (smallest face size 105x128 pixels), (c) neutral emotion for both dancers (smallest face size 95x122 pixels). Image (f) is the closest to the viewing camera, image (c) the farthest. Images from the PREMIERE Dance Motion Dataset. As comparison, images size in the Talawa video is of 1280x720 pixels and the size of dancer faces detected in Figure 12 is around 83x89 pixels.

2.3.4. Pose estimation refinement

The fourth task to analyse a video content (Task T9 of the image processing pipeline illustrated in Figure 1) is to refine the accuracy of the output data resulting from Task 7 by removing low confidence score, smoothing 2D pose estimation over time (temporal filtering) using Kalman filter, and other efficient spatio-temporal filtering technics such as SmoothNet22 or HANet23.

- Input: individual scenes (format H264 video or JSON, and Python binary format - pickle) (output from Task 7) with for each frame a set of keypoints for each person, and a 2D bounding box (with 2D position).
- Output: individual scenes (format H264 video or JSON, and Python binary format - pickle) with for each frame a set of keypoints for each person, and a 2D bounding box (with 2D position).

2.3.5. Laban Movement Analysis

A fifth task (Task T10 of the image processing pipeline illustrated in Figure 1, connected to Task 3.2 of the WP3), related to the Laban Movement Analysis (LMA), could potentially be added to the 2D video content analysis. Laban's method24, with other techniques taught in dance schools, enables us to understand, describe, interpret and document how humans move their body. It exists in the State-Of-the-Arts very few estimation models of Laban features from skeleton, among them we can cite [14]. By analyzing the posture, the alignment of joints, the angles of joints, the rotations of joints, the distance between the joints, etc. over time, these models can estimate human body movement features.

2.4. 3D human body pose and shape estimation

The objective of this task is to automatically build a 3D human body and shape mesh model from a single 2D pose model or a set of 2D poses (Task T11 of the image processing pipeline illustrated in Figure 1). It exists in the State-Of-the-Art efficient methods to compute a 3D mesh model from 2D pose, such as HMR 2.025, PyMAF-X26, OSX27, etc. If in one video sequence, we have only access to the upper body of dancers (as in Figure 13) then it is better to use the OSX mesh recovery method. OSX representation is based on SMPL-X.

- Input: individual scenes (format H264 video or JSON, and Python binary format - pickle) with for each frame a set of 2D keypoints for each person, and a 2D bounding box (with 2D position) (output from Task 9).
- Output: individual scenes (format H264 video or JSON, and Python binary format - pickle) with for each frame and for each people: a 3D model defined by SMPL/SMPL-X parameter values28 and a set of 3D keypoints, and a 3D bounding box (with 3D position).

2.5. 3D human body tracking

The objective of this task is to automatically track each people in movement in a video sequence even if part of his/her body is occluded by another people or object (Task T12 of the image processing pipeline illustrated in Figure 1). It exists in the State-Of-the-Art efficient methods to track people in a video sequence from 3D observations over time, such as PHALP29. Tracking people in video sequences is also useful to improve the accuracy of 3D pose estimation (to smooth short temporal drifts as in Figure 19 (a) or to smooth short

temporal incoherencies in 3D pose estimation results, as in Figure 19 (c)). To improve the accuracy of 3D pose estimation few other tracking techniques could be used. For example, as the two dancers of the PREMIERE dance motion dataset wear both dark suits, when complex occlusions occur (as in Figure 3) it is very challenging to trach each of them, except if we focus on face features.

- Input: individual scenes (format H264 video or JSON, and Python binary format - pickle) with for each frame and for each people: a 3D model defined by SMPL/SMPL-X parameter values and a set of 3D keypoints, and a 3D bounding box (with 3D position) (output from Task 11).
- Output: individual scenes (format H264 video or JSON, and Python binary format - pickle) with for each frame and for each people: a 3D model defined by SMPL/SMPL-X parameter values, a set of 3D keypoints, a 3D bounding box (with 3D position) and an attribute (for example, this attribute is display in color in Figures 18 and 19).

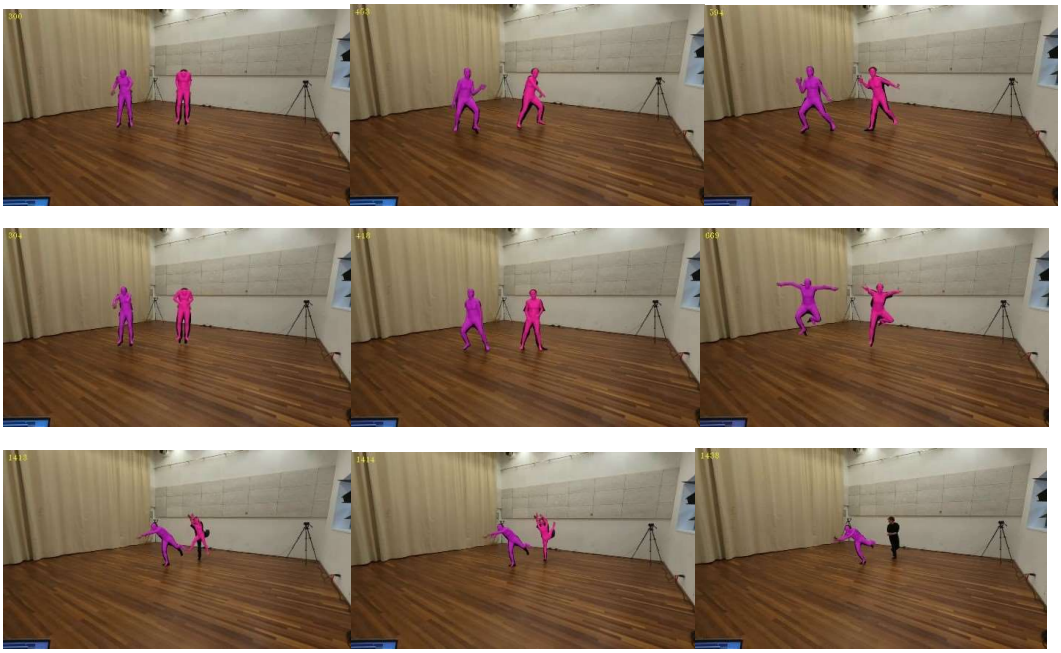
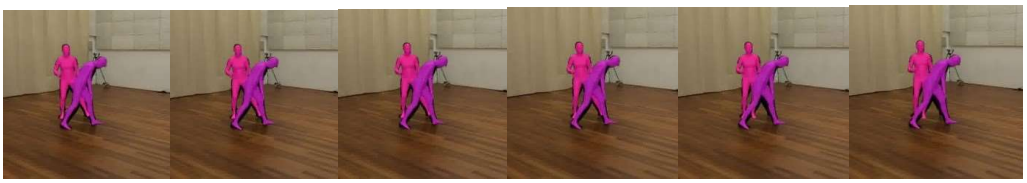
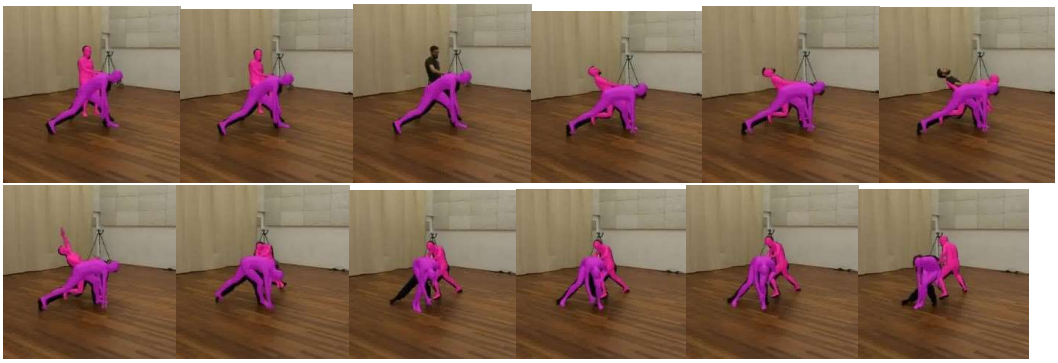


Figure 18 : Examples of inaccurate 3D meshes obtained using the PHALP method. Images from the PREMIERE Dance Motion Dataset. On top images: hands are not well positioned (due to the blur related to high-speed hand motion, in such cases confidences scores are low). On second row: feet are not well oriented (due to geometrical constraints defined for angle of joints, more flexibility should be given to dance videos). On bottom images: the meshing is completely loss (due to low accuracy of 2D pose estimations, that concerns very few consecutive frames in this video). For most of the frames of this video sequence, the shape and the size of meshes fit well with human bodies

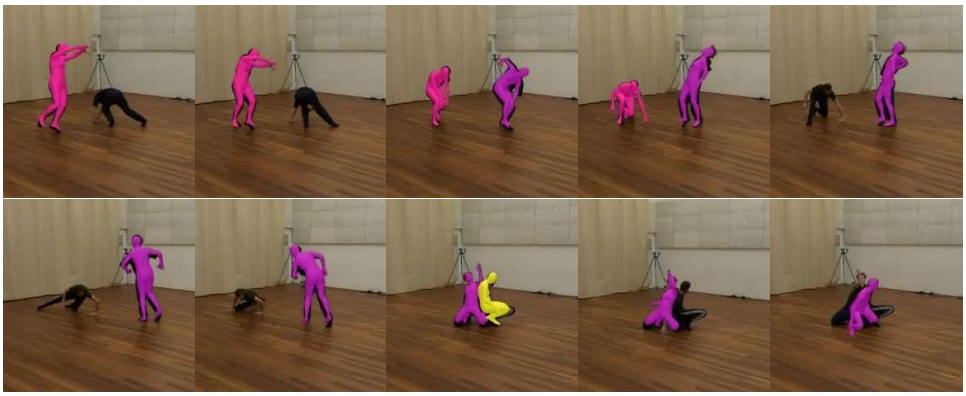
frames in this video). For most of the frames of this video sequence, the shape and the size of meshes fit well with human bodies.



(a) Zoom on image frames 63, 64, 65, 66, 67, 68



(b) Zoom on image frames 98, 99, 100, 128, 138, 143, 154, 225, 239, 240, 242, 253.



(c) Zoom on image frames 634, 646, 666, 689, 694, 712, 729, 1088, 1111, 1117.

Figure 19: Examples of inaccurate 3D meshes obtained using the PHALP method. Images from the PREMIERE Dance Motion Dataset. (a) In this challenging video sequence: the tracking is progressively lost due to a higher occlusion between legs of the dancers. (b) In another sequence of this video: tracking results are incoherent from one frame to another one due to a strong occlusion between dancers (this occlusion problem can be alleviated from the video captured with the background camera on the right side of the scene). (c) In another sequence of this video: tracking results are unstable due to the position of one dancer on the ground (pose estimation generates very low confidence scores).

2.6. Trajectories estimation and Actions parsing

The objective is to estimate the trajectory of each people moving in a video sequence from their 3D position over time, and to analyse the actions they did during the video sequence in which they appeared (Task T13 of the image processing pipeline illustrated in Figure 1). To reach this objective, the first task is to recognize people in a video, in some case tracking people in not sufficient (e.g. when in a video sequence a dancer moves out, next moves in, the visual field of the camera). It exists in in the State-Of-the-Art very few trajectory estimation methods, we can nevertheless cite the TriPOD method [15]. To the best of our knowledge, actions parsing from 3D body pose estimation and tracking was not yet being fully investigated30.

2.7. Semi-automatic analysis

The various tests and analyses we performed to define this processing pipeline have been based on a very diverse set of videos with highly complex content, selected by members of the PREMIERE consortium. Despite the very conclusive results we obtained, we concluded

that it would be very difficult, if not impossible, to define unique parameters for all the tasks of this pipeline that would work on all videos. Therefore, it will be necessary to create software tools that, with the help of a human expert, could contribute to validate and consolidate the results obtained. This will be one of the tasks that UJM will address in the next version of the deliverable.

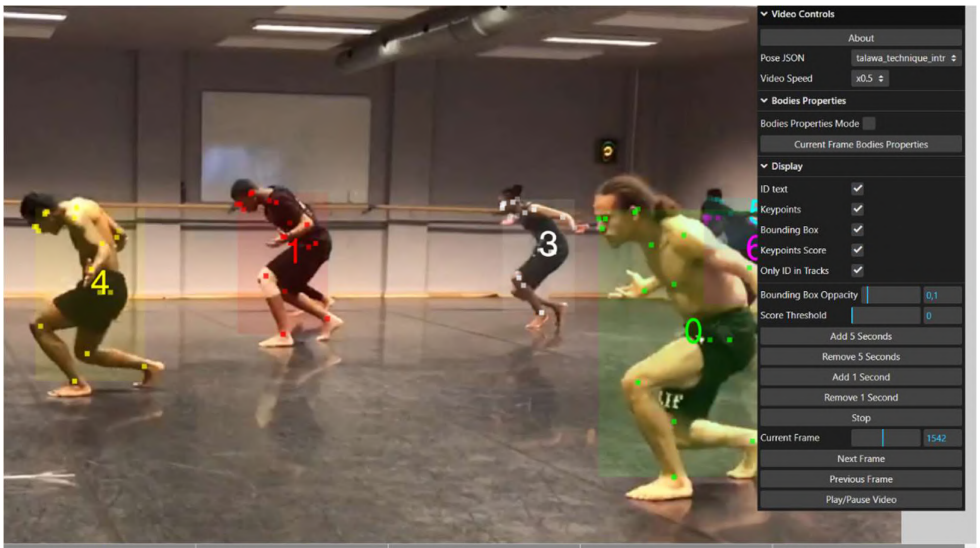
3. Internal tools and software engineering

In the many studies we have conducted, we have also concluded that it would be necessary to develop visualization and annotation tools needed to perform the tasks of this work package. We did various tests and experiments to select the most relevant platforms (visualization and annotation tools) needed to perform the tasks of this work package.

3.1. Internal tools

To meet the requirements of the PREMIERE project, we have chosen to develop all our visualization and annotation tools using only web technologies (HTML, Javascript, WebGL, etc.). They can be run on a web server or from a Python script.

The first tool is dedicated to the visualization and comparison of 2D pose detections based on the keypoints obtained (see Figure 20).

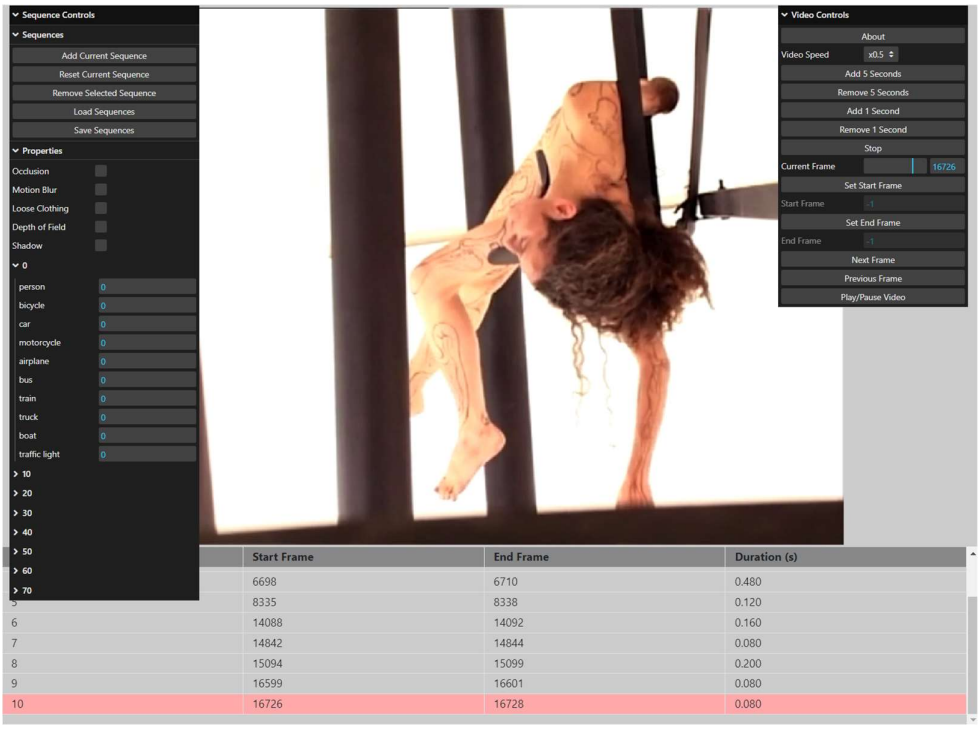


Body Track # (116)	Body ID	Start Frame	End Frame	Duration (s)
0	0	101	254	6.120
1	1	101	254	6.120
2	0	256	271	0.600
3	1	256	342	3.440
4	2	256	271	0.600
5	0	273	342	2.760
6	2	322	342	0.800

Figure 20: : Web server platform for visualization and comparison of 2D pose detection results. Image from the Talawa video

Two other tools have been developed to annotate and validate our results:

- Sequence annotation of frame content: number of people present, objects present, presence of occlusions, motion blur, etc (see Figure 21).



	Start Frame	End Frame	Duration (s)
	6698	6710	0.480
5	8335	8338	0.120
6	14088	14092	0.160
7	14842	14844	0.080
8	15094	15099	0.200
9	16599	16601	0.080
10	16726	16728	0.080

Figure 21: Web server platform for sequences annotation

- Validate the bounding box used for tracking people in a sequence

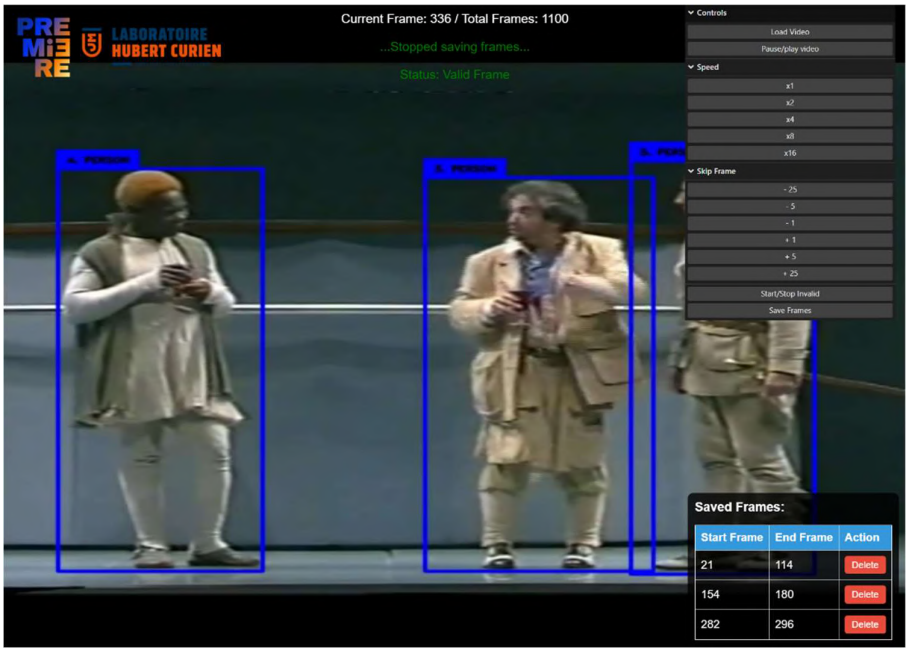


Figure 22: Web server platform for validation of bounding boxes

3.2. Software engineering

A computer or virtual machine with the following minimum characteristics is required to process the tasks of this pipeline:

- NVIDIA graphics cards with at least 16 GB of VRAM (based on the Ampere or Ada Lovelace architecture)
- 8 cores 64 bits CPU
- 32 GB of RAM

The target operating system is Ubuntu 22.04 LTS, which can be run natively or through WSL (if people want to use Windows 10 or 11 Pro).

4. Conclusions

In conclusion, this deliverable provides an overview of our approach to leverage Task T4.1 - 3D scene Analysis and Understanding, and Task T4.2 - 3D pose trajectories estimation in complex scenes. Our work encompasses the development of a video processing pipeline combining cutting-edge deep learning architectures, models and conventional video processing algorithms and tools. Throughout our research, we aim to push the boundaries of 3D analysis and understanding methods, and 3D pose trajectories estimation method, particularly in the context of dance, theatre data, and complex scenes.

In the next six months (i.e. month M21 of the project), our research efforts will focus on improving: - the robustness of our pipeline against occlusion, motion blur, frame rate, noise, etc.; - the efficiency and the accuracy of our pipeline when applied to contemporary dance and archives. We will investigate solutions that leverage the limits of existing approaches. We will also investigate solutions for end-users to refine the results obtained. To achieve this goal, we will delve into cutting-edge deep learning architectures and conventional video processing algorithms. This will require also to setup efficient accuracy metrics.

References

[1] Ruilong Li, Shan Yang, David A. Ross, Angjoo Kanazawa, *AI Choreographer: Music Conditioned 3D Dance Generation with AIST++*, https://arxiv.org/pdf/2101.08779.pdf

[2] Shuhei Tsuchida, Satoru Fukayama, Masahiro Hamasaki, Masataka Goto, *AIST dance video database: multi-genre, multi-dancer, and multi-camera database for dance information processing*, https://archives.ismir.net/ismir2019/paper/000060.pdf

[3] Zhengqin Li, Jia Shi, Sai Bi, Rui Zhu, Kalyan Sunkavalli, Milos Hasan2, Zexiang Xu, Ravi Ramamoorthi, and Manmohan Chandraker, *Physically-Based Editing of Indoor Scene Lighting from a Single Image*, https://arxiv.org/pdf/2205.09343.pdf

[4] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, Yong Jae Lee, *Segment Everything Everywhere All at Once*, https://arxiv.org/abs/2304.06718

[5] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, Fisher Yu, *Segment Anything in High Quality*, arxiv.org/pdf/2306.01567.pdf

[6] Dipika Gupta, Manish Kumar, Sachin Chaudhary, A systematic review of deep learning frameworks for moving object segmentation, https://link.springer.com/article/10.1007/s11042-023-16417-3

[7] Yizhe Zhang, Shubhankar Borse, Hong Cai, Fatih Porikli, AuxAdapt: Stable and Efficient Test-Time Adaptation for Temporally Consistent Video Semantic Segmentation

[8] Hyojin Park, Alan Yessenbayev, Tushar Singhal, Navin Kumar Adhikari et al., Real-Time, Accurate, and Consistent Video Semantic Segmentation via Unsupervised Adaptation and Cross-Unit Deployment on Mobile Device,

[9] Rawal Khirodkar, Visesh Chari, Amit Agrawal, Ambrish Tyagi, Multi-Instance Pose Networks: Rethinking Top-Down Pose Estimation,, https://arxiv.org/pdf/2101.11223.pdf

[10] Haoming Chen, Runyang Feng, Sifan Wu, Hao Xu, Fengcheng Zhou, Zhenguang Liu, 2D Human Pose Estimation: A Survey, https://arxiv.org/pdf/2204.07370.pdf

[11] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, Bernt Schiele, 2D Human Pose Estimation: New Benchmark and State of the Art Analysis,

[12] Chenyan Wu, Yukun Chen, Jiajia Luo, Che-Chun Su, Anuja Dawane et al., MEBOW: Monocular Estimation of Body Orientation In the Wild,

[13] Yanlei Gu, Huiyang Zhang, Shunsuke Kamijo Multi-Person Pose Estimation using an Orientation and Occlusion Aware Deep Learning Network, https://www.mdpi.com/1424-8220/20/6/1593.

[14] Ziya Erko¸ Serkan Demirci, Sinan Sonlu, Ugur Gudukbay, Skeleton-based Personality Recognition using Laban Movement Analysis,,

[15] Vida Adeli, Mahsa Ehsanpour, Ian Reid, Juan Carlos Niebles, Silvio Savarese, Ehsan Adeli, Hamid Rezatofighi, TRiPOD: Human Trajectory and Pose Dynamics Forecasting in the Wild, https://arxiv.org/abs/2104.04029