care, judgment, dexterity

CRAEFT

Scene and activity monitoring

Project Acronym	Craeft
Project Title	Craft Understanding, Education, Training, and Preservation for Posterity and Prosperity
Project Number	101094349
Deliverable Number	D3.3
Deliverable Title	Scene and activity monitoring
Work Package	29
Authors	Gavriela Senteri, Sotiris Manitsaris

This project has received funding from the European Commission, under the Horizon Europe research and innovation programme, Grant Agreement No 101094349.
http://www.craeft.eu/

Executive summary

This deliverable analyses the actions taken for scene understanding and activity monitoring in the context of traditional crafts. The Introduction highlights the complexity of traditional crafts and the dexterity required, making preservation efforts particularly challenging. It emphasizes the role of machine learning (ML) in analysing and modelling these skills using data captured through video, sound, and other sensors as a crucial development in this field.

The Motion Capture section provides an overview of the dataset created within the project, encompassing a variety of craft professions such as glassblowing with pipe, etc. Data was collected from different perspectives and modalities to ensure that all critical aspects of the crafting process were captured.

In Scene Understanding, the focus is on how Neural Radiance Fields (NeRFs) and 3D Gaussian Splatting (3DGS) technologies are employed to reconstruct 3D scenes from 2D video data, effectively overcoming the limitations of traditional scene analysis methods. This section also discusses the data processing methods used and analyses the performance of both scene representation techniques.

The Activity Monitoring section explores the challenges of recognizing and analysing human gestures within crafts. It discusses advanced ML methods such as Meta-Learning (MetaL) and Multi-Task Learning (MultiTL). Additionally, it evaluates the performance of a Multi-Task Learning (MTL) module about its contribution to gesture recognition using biomechanical primitives.

In the Conclusion, the findings highlight the significant potential of these ML techniques in advancing cultural heritage preservation and vocational training. Future work will focus on refining these models, particularly in enhancing the representation of dynamic actions and ensuring adaptability to new, unseen crafts with minimal retraining.

Document history

Date	Author	Affiliation	Comment
15/07/2024	Gavriela Senteri	ARMINES	First draft
24/8/2024	Xenophon Zabulis	FORTH	Formatted
27/8/2024	Sotiris Manitsaris	ARMINES	Executive abstract
28/8/2024	Xenophon Zabulis	FORTH	Formatted, grammar corrections

Abbreviations

ML	Machine Learning
NeRFs	Neural Radiance Field
3DGS	3D Gaussian Splatting
ICH	Intangible Cultural Heritage

MetaL	Meta-Learning
MTL	Multi-Task Learning
MLP	Multi-Layer Perceptron
DL	Deep Learning
SfM	Structure from Motion

Table of contents

1. Introduction

The preservation and transmission of traditional crafts, as well as the vocational training of craftspeople, are major challenges in the field of cultural heritage. However, crafts often involve complex techniques, that require specific skills and a level of dexterity, making preservation efforts particularly difficult. Recent advances in ML, facilitate this preservation, making possible the analysis and modelling of these skills, using data captured through video, sound and other modalities and sensors. The combination of sensor technology and ML has created new opportunities in the field of crafts by enabling the recognition and interpretation of human activities. This is a critical point for improving the vocational training of craftspeople that require a specific level of dexterity, as well as the digitisation and further understanding of ICH.

The scene of a craftsman can be specifically separated into two main components, the main scene includes the workbench, along with all the used materials and tools. The second component concerns the human movement, thus the activities of the craftsman, their analysis and recognition. In this deliverable, we will focus on both components, where different ML methods are used both for the understanding of the main working scene, and the recognition and further analysis of the craft movements.

Initially focusing on scene understanding, current scene analysis technology is limited to 2D approaches, that primarily analyse the respective craft scene through video data. The limitations of 2D analysis prevent us from capturing enough detail to fully understand the craft process and guide craft beginners in mastering complex craft skills. Consequently, there is a need to extend scene analysis to 3D scenarios to better capture and understand the craft scene components. In recent years, Radiance Field Rendering (specifically methods such as the NeRFs and the 3DGS) has made significant progress in generating realistic, both static and dynamic 3D scenes. However, most of this research has focused on third-person view-based rendering and synthesis. The use of this view for scene rendering poses challenges in capturing the intricate motion details of craftsmen due to occlusions in the working environment. In contrast, egocentric vision offers a promising approach for observing and analysing a craftsman's manipulation in complex crafting processes. Consequently, our work uses egocentric video to capture the craft scene and construct a dynamic 3D scene that enhances the understanding of the crafting process and provides guidance for novices to follow expert actions.

A reasonable question would concern the difference between a simple video of the craft process and the reconstructed video as the result of the ML methods mentioned above. A video can indeed provide the needed details, and capture the colours and the utilized tools and materials, however, it is limited to the single perspective from which it was captured. On the other side, the Radiance Field Rendering methods provide a full 3D reconstruction of the scene, allowing users to perceive and analyse the depth and spatial arrangement of the crafting process. Apart from this, they also enable the interactive exploration of the scene. Users can zoom in, rotate, and view the scene from different angles, enhancing their understanding of complex actions. Lastly, these methods allow for detailed inspection of specific parts of the scene, such as intricate hand movements or material deformation, and can be used to simulate different craft scenarios, providing a versatile tool for training and experimentation. In this work, both the NeRFs and the 3DGS methods were deployed, to test the results that they can provide.

Moving on to the activity monitoring part of the craft environment, ML modules can support or assist craftsmen during their routine or even guide them in the process of training for a craft. The use of action recognition has the potential to convey and conserve the complex skills required in UNESCO-listed occupations, such as marble carving and glass blowing. However, movement recognition in professional environments faces significant scientific challenges, primarily due to the complexity of data and the need for ML methods to deliver accurate, robust results.

The challenges concern 1) the complexity of human motion data, 2) the transferability and generalization capabilities of the implemented algorithms, and 3) how an action is interpreted through ML. Focusing on data complexity, and the variability of data involved in action recognition is the first significant challenge [2]. Human actions consist of a wide range of movements, that can be captured from different angles, or through different data modalities resulting in complex data representations. Managing this level of complexity requires the design of effective feature extraction techniques that do not take into consideration redundant or irrelevant information. Furthermore, most ML implementations, require a vast amount of training data, as such, robust algorithms able to decompose human movement data into simpler movement representations are essential. The second major challenge of transferability and generalization concerns the ability of algorithms to properly utilize prior knowledge, and generalize to apply to more than one training dataset at a time. It is observed that due to variations in lighting conditions, background, as well as individual variations in the execution of the same human movement, models trained on a single dataset may not perform well when applied to new scenarios or environments [2]. Thus, to achieve the adaptability of action recognition models, effective domain adaptation techniques, and methods for learning invariant representations able to generalize well across diverse datasets and conditions are essential. Other fundamental challenges in action recognition and human motion analysis concern the tasks of motion prediction and interpretation. Human motion is dynamic and can differ considerably between individuals, making it difficult to accurately predict and recognize gestures [3]. This difficulty involves developing models that can account for the individual variation in the anthropometric characteristics, environmental conditions, and other factors that can affect the understanding of the gesture's intended meaning. In addition, it is essential to address temporal dependencies to enhance the accuracy of motion prediction and interpretation.

The above scientific challenges, come along with the increasing need of the gesture recognition community, for solutions that can accommodate the subtle intricacies of human movements, leading to advances in ML beyond what has been traditionally implemented. In this regard, this community is interested in building technologies capable of precisely interpreting fine details of human gestures in real-time; an attribute vital for several applications. Such applications call for high precision, low latency gesture recognition, and flexibility toward any user or context without full retraining. Additionally, there is a growing demand for their robustness when factors such as changing light conditions, and occlusions affect their performance. This means that existing ML models cannot be used beyond certain limits, and thus new methods have to be developed that can handle the above needs. Methods that appear promising in the literature, for handling the aforementioned challenges, are Meta-Learning (MetaL) and Multi-Task Learning (MultiTL). They both tackle the problems of data complexity, generalization, and transferability, through exchanging knowledge among related tasks, processed in parallel for MultiTL, and by leveraging existing knowledge, for the adaptation to new, unseen tasks, for MetaL. However, the current literature reviews on MetaL, and MultiTL, focus on applications within the general framework of Deep Learning, overlooking their potential in specific fields, such as human movement recognition.

This deliverable is structured in two main parts that concern scene understanding and activity monitoring. In the first part, the methods of NeRFs and 3DGS are presented, followed by the experiments implemented and the qualitative, as well as quantitative results on video reconstruction with both methods. In the second part, which focuses on activity recognition/monitoring, the proposal of a hierarchical structure that decomposes the human movement in crafts is presented, along with the used methodology, as well as the recognition results with the deployed method of MTL.

2. Motion capture

Within the framework of this project, a dataset was created with a variety of craft professions, and pilots of CRAEFT, including glassblowing with pipe, glassblowing with blowtorch, marble carving, silversmithing, and porcelain pottery. The recordings took place in the first year of the project at the respective craft environment, in collaboration with experts in each craft, that are either responsible for teaching a craft, or for creating and promoting their craft through an association, or the family business.

Concerning the technical part of the recordings, to make sure that all the important details of the creation process would be captured, different sensors were deployed. The goal is by the end of the project to further explore the complementarity of these sensors and the captured data.

More specifically on the used sensors, two GoPro cameras were used, one in an egocentric (first-person) view, and one in an exocentric (third-person view). The egocentric view provided the dexterous hand movements of the craft operators, along with the used tools, while the exocentric view captured the operator's ample body movements.

Furthermore, two microphones, a contact one that captured the sound of the interaction between tools and materials, through their vibrations, along with a stereo microphone that recorded the sounds from the working environment, work objects, and materials, as well as the communication among different operators in collaboration, creating a multi-modal setup.

Table 1: Examples from the egocentric (top images) and the exocentric (bottom images) of the craft motion recordings performed for each one of the CRAEFT pilots in real craft laboratories.

Glassblowing with pipe	Glassblowing with blowtorch	Marble carving
		
		

Silversmithing

Porcelain pottery

3. Scene understanding

This section briefly explains the radiance field, a fundamental concept in scene rendering, through two key methods, the Neural Radiance Fields (NeRFs) and the 3D Gaussian Splatting (3DGS), moving forward right after the presentation and analysis of the extracted results.

3.1. Neural Radiance Field

In computer graphics, light is modelled as a continuous phenomenon that travels through space in straight lines, each carrying specific colours and intensities. This concept forms the basis for understanding how light interacts with objects in a scene to produce the images we see. A radiance field represents the amount of light travelling in every possible direction through every point in space. Thus, the radiance field can be defined as follows:

$$L: R^3 \times S^2 \rightarrow R^3$$

where R^3 represents the position (x, y, z) in spatial coordinates, S^2 represents the directions (θ, ϕ) in spherical coordinates, and the output is a radiance value in R^3. In computer graphics, the function L can be expressed through implicit or explicit representation, with each method having its specific advantages for scene representation and rendering.

The implicit radiant field represents a continuous volumetric light distribution of a scene without relying on explicit geometric information. It uses a neural network to approximate the light distribution across the continuous volume [6]. In 2020, Mildenhall et al. proposed NeRF (Neural Radiance Fields) [7], which uses a Multi-Layer Perceptron (MLP) to map the light radiance field into a colour field (R, G, B) and a density field σ.

Simpler, NeRFs use Deep Learning (DL) to create a 3D representation of a scene from a few 2D images. Using a practical example, in the case where several photos of an object were taken from different angles, NeRFs can take these photos and use them to build a detailed 3D model of that object, capturing all its details and textures. This way images are turned from flat, into rich, three-dimensional representations.

In 2021, NeuralDiff [10] extended NeRFs to dynamic egocentric video rendering by splitting the dynamic scene into a static foreground, dynamic background, and actor. It uses three side-by-side MLPs to render each scene separately and capture human motion.

NeuralDiff [10] is the first approach in utilizing egocentric video to render an Egocentric Dynamic Radiance Field. The core idea of NeuralDiff is that a dynamic scene viewed at a time step t , x_t can be modelled as a function: $x_t = f(B, F_t, g_t)$ where B represents the static background, $F_t$denotes the variable foreground at time t, and g_t is the camera view direction at this time step. Based on this idea, neural rendering techniques predict novel views of static objects under varying viewpoints and dynamic foreground objects. Then based on this framework, we can import the time step feature to the MLP to render the dynamic foreground object. Consequently, the NeRF can generate the different parts of the

scene at the same view, then we combine these different parts as one image comparing with our ground truth to learn the NeRF parameters.

3.2. 3D Gaussian splatting

The 3D Gaussian Splatting [8] is an explicit rendering method that uses point clouds and 3D Gaussians to represent the geometric scene. Spherical harmonic functions can then model the interaction of light with the environment, capturing reflections, materials, textures, and all the visual appearances of 3D Gaussian elements.

The deformable 3D Gaussians [9] is a variant of 3D Gaussian Splatting. Unlike the original 3D Gaussian Splatting, which places geometric elements under the world coordinate system, Deformable-3DGS places the 3D Gaussians in canonical space. It then uses an MLP network to predict the offset $(\delta x, \delta y, \delta z)$ of each 3D Gaussian at each time step in the canonical space, using the encoded position information $\gamma(sg(x))$ and time information $\gamma(t)$ as the input of the network. Based on this idea, the deformable 3D-GS can generate novel views of a dynamic scene while maintaining the real-time rendering property of 3D Gaussian Splatting.

Again, more simply, the 3D Gaussian Splatting model takes a different approach from the NeRFs. It represents the scene using a grid of points, each described by a simple mathematical shape called a Gaussian. An example would be to think of 3DGS as a method that creates a 3D picture using a lot of tiny, colourful pixels that blend smoothly to form the final image. This method helps in efficiently organizing and rendering the 3D scene, making it look realistic and detailed.

3.3. Deployed methods and results

Since both the NeRF method, as well as the 3D Gaussian Splatting appear promising in the field of scene rendering, for scene understanding, these two methods were separately deployed, to further examine the results that they provide. As such, in this section, the focus will be mainly given on Neural-Diff, the extension of the NeRF method, and the 3DGS.

To illustrate the performance of NeuralDiff and Deformable-3DGS in a complex, dynamic scene-reconstruction case, we selected two of the captured craft scenarios, marble carving and glassblowing, before further proceeding to the rest. The two pilots were initially chosen, as the marble carving use case shows sequential hand and material rotations that need to be translated in every period, while the glassblowing features transparent materials, deformation processes, and flames, along with a constant rotation of the human hand. This second case presents a significant challenge due to the incorporation of numerous complex elements.

Figures 1 and 2 show the pipelines we used in our study. Both, NeuralDiff and the Deformable-3DGS were deployed to reconstruct dynamic scenes.

3.3.1 Data processing

Concerning the processing of the captured data, accurate camera pose can improve 3D reconstruction

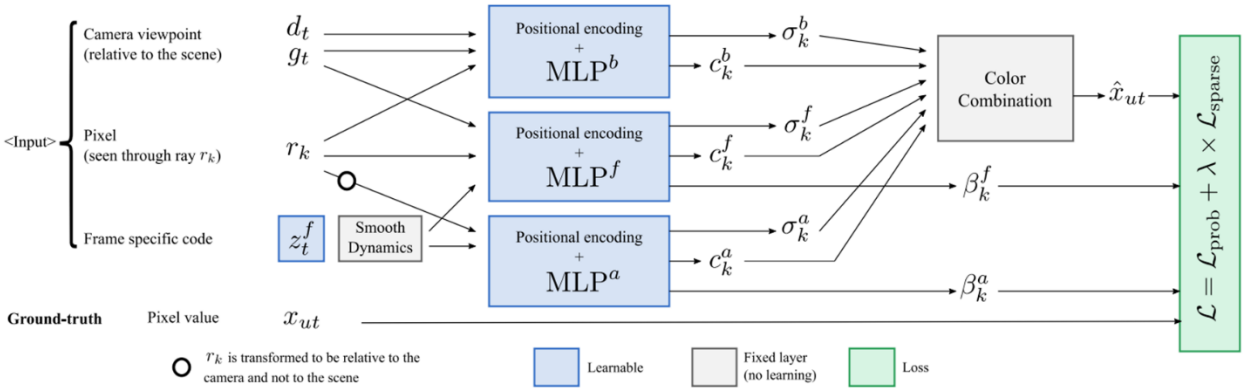

Figure 1: Network structure of NeuralDiff [10]. The architecture uses three parallel MLP layers to capture different elements: one for the static background, and two for dynamic foregrounds (one for dynamic objects in the scene, and one for the human hand, referred to as the actor). The first MLP layer utilizes camera viewpoint information and ray information. The second and third layers incorporate time-coding features and pose information in the world frame for dynamic objects, and the camera frame for the actor, to predict the colour value of a pixel.

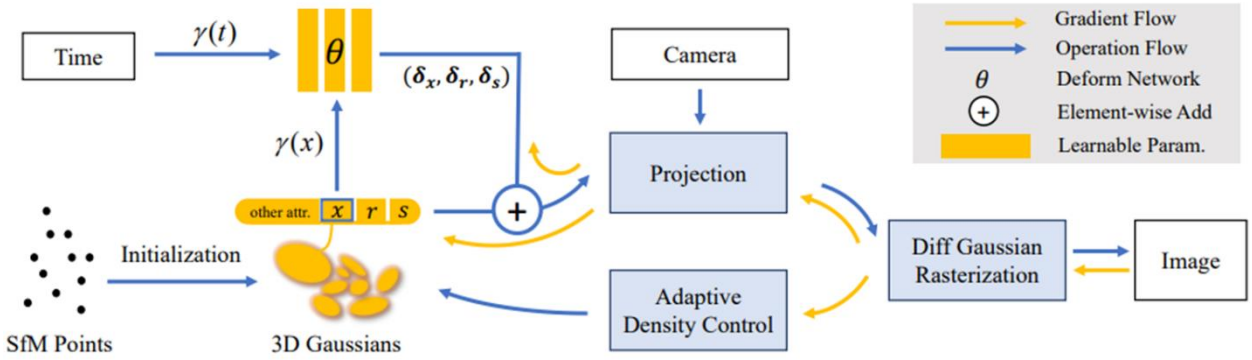

Figure 2: Deformable 3D Gaussian Splatting Pipeline [9]. Details extracted from the COLMAP are used as the initialization of 3D Gaussian, combining an MLP as the deformable network predicts the offset of dynamic 3D gaussian in canonical space based on position information. During the initial 3000 iterations, only the density of the 3D Gaussians and their parameters are updated. After the 3000th iteration, both the deformable network and the 3D Gaussian parameters are updated.

results. As such, in this case, the camera poses of each craft egocentric video frame are estimated with the use of COLMAP, a popular Structure from Motion (SfM) software. COLMAP reconstructs 3D structures from a set of 2D images by estimating the camera positions and orientations (poses) and generating sparse or dense point clouds. The use of camera poses in NeRF and 3D-GS methods differs significantly. As an implicit rendering method, NeRF (Neural Radiance Fields) samples spatial points along rays computed from the estimated poses. It then employs a multi-layer perceptron (MLP) to learn the colour and density parameters of each sampled point. Once trained, the MLP provides the colour and density information needed to compute the final pixel value of these rays, effectively performing a backward mapping. In contrast, the 3D-GS (3D Gaussian Splatting) method reverses this process. It starts by obtaining a sparse point cloud from COLMAP and initializes 3D Gaussians based on this sparse point cloud. These 3D Gaussians are then projected into image space and rasterized in parallel, performing a forward mapping.

3.3.2 Experiments and Results

We run two baselines on our collected craftsman egocentric video data. All of the experiments were done on an NVIDIA RTX 3060.

- The NeuralDiff model is trained for 10 epochs with a learning rate of 0.0005, using a 228 x 128 image as input.
- The Deformable 3D Gaussian Splatting model was also trained with the same size image 228 x times 128, with 40,000 iterations. The first 3,000 iterations are the warm-up step to densify the 3D Gaussian (Figure 5).

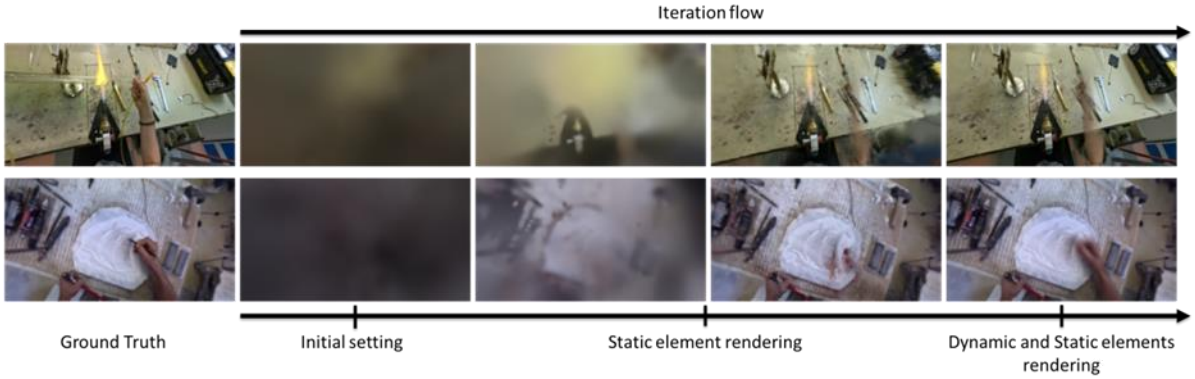

Figure 1:Defromable 3D Gaussian Splatting visualization results on 3 different craft processes. It is observed that during the first 3000 iterations, static objects are rendered, while after the 3000th iteration, both static and dynamic objects are reconstructed.

3.3.2.1 Quantitative results

For each craft case, a 1-minute video clip at 30 FPS was used, with a frame step of 2, concluding with the use of 900 training frames. Since the fast motion in the egocentric video causes a large loss of information for dynamic objects, a much denser concentration of training frames is used than normal.

To assess the performance of our model, the Peak Signal-to-Noise Ratio (PSNR), the structural similarity index (SSIM), and the VGG-based perceptual similarity metric (LPIPS) are deployed.

- PSNR (Peak Signal-to-Noise Ratio): This method quantifies the quality of reconstructed images by evaluating pixel-wise differences between the original and reconstructed images. A high PSNR indicates that the Mean Squared Error (MSE) between the two images is small, meaning the reconstructed image quality is better.
- SSIM (Structural Similarity Index): Unlike PSNR, SSIM does not directly rely on pixel error. Instead, it samples pixels using windows, assessing the local mean, local standard deviation, and local covariance of pixels to evaluate image similarity. SSIM values range from -1 to 1, where 1 indicates perfect similarity between the original and reconstructed images.
- LPIPS (Learned Perceptual Image Patch Similarity): LPIPS uses a pre-trained neural network to obtain feature representations and then calculates the distance between the feature maps of the two images. Lower LPIPS values indicate higher similarity. LPIPS is particularly useful because it captures perceptual differences that align with human visual perception.

In this deliverable, our primary goal was to reconstruct the entire dynamic scene, focusing particularly on the dynamic foreground and the movement of the human hand at each time step. To evaluate the performance of our models, we directly compared the rendered images at each step with the ground truth images. Additionally, we assessed training time and model rendering performance to determine the feasibility of using the model in real-time applications.

Table 2: Presentation of the computed PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), and LPIPS (Learned Perceptual Image Patch Similarity) metrics for two different craft scenarios: marble carving and glass blowing. The two models being compared are Deformable 3D Gaussian Splatting (Defor-3DGS) and NeuralDiff.

Method	**Marble carving**			**Glass blowing**		
	PSNR	SSIM	PIPS	PSNR	SSIM	PIPS
Defor-3DGS	29.15	0.91	0.12	28.86	0.92	0.11
NeuralDiff	23.74	0.89	0.16	21.44	0.85	0.22

Table 3: In the context of FPS and Training time, Deformable 3D Gaussian Splatting generally surpasses NeuralDiff across various scenarios, thus Deformable-3D Gaussian Splatting can realize a real-time rendering

Method	**Marble carving**		**Glass blowing**	
	Training time	Frames per second	Training time	Frames per second
Defor-3DGS	1h35	30	1h32	30
NeuralDiff	10h12	2	10h27	2

The Defor-3DGS model demonstrates superior performance in both the PSNR and SSIM metrics across the scenarios of marble carving and glassblowing, indicating better overall image quality with lower reconstruction error and greater structural similarity between the rendered and ground truth images compared to NeuralDiff. Additionally, Defor-3DGS achieves lower LPIPS values, suggesting that it captures perceptual differences more effectively, producing images that are more visually similar to human perception. These results consistently show that Defor-3DGS provides higher-quality reconstructions even as scene complexity increases, maintaining a performance edge over NeuralDiff despite the greater challenge posed by more intricate and dynamic scenes.

In terms of training efficiency and real-time rendering capability, Defor-3DGS requires significantly less training time than NeuralDiff in both evaluated scenarios. This efficiency is crucial for applications where rapid model deployment or frequent updates are needed. Moreover, the Defor-3DGS model achieves a rendering speed of 30 frames per second (FPS), which qualifies as real-time performance, making it suitable for dynamic and interactive applications. In contrast, NeuralDiff's rendering speed of 2 FPS is far below real-time, indicating that it struggles to process dynamic scenes quickly enough for real-time use.

Overall, the results strongly suggest that Deformable 3D Gaussian Splatting (Defor-3DGS) is superior to NeuralDiff in both image quality and real-time rendering performance. The ability of Defor-3DGS to efficiently render high-quality images in real time makes it a viable solution for dynamic scene reconstruction, particularly in applications where real-time processing is critical, such as virtual reality, augmented reality, and real-time 3D graphics in games and simulations.

3.3.2.2 Qualitative results

In this section, we will conduct a qualitative analysis of each model for both marble carving and glass blowing. We will begin by examining the output of each model to evaluate their overall performance. Following this, we will analyze the rendered images in the context of dynamic scenes, focusing on how well each model reconstructs dynamic objects within the craft environment.

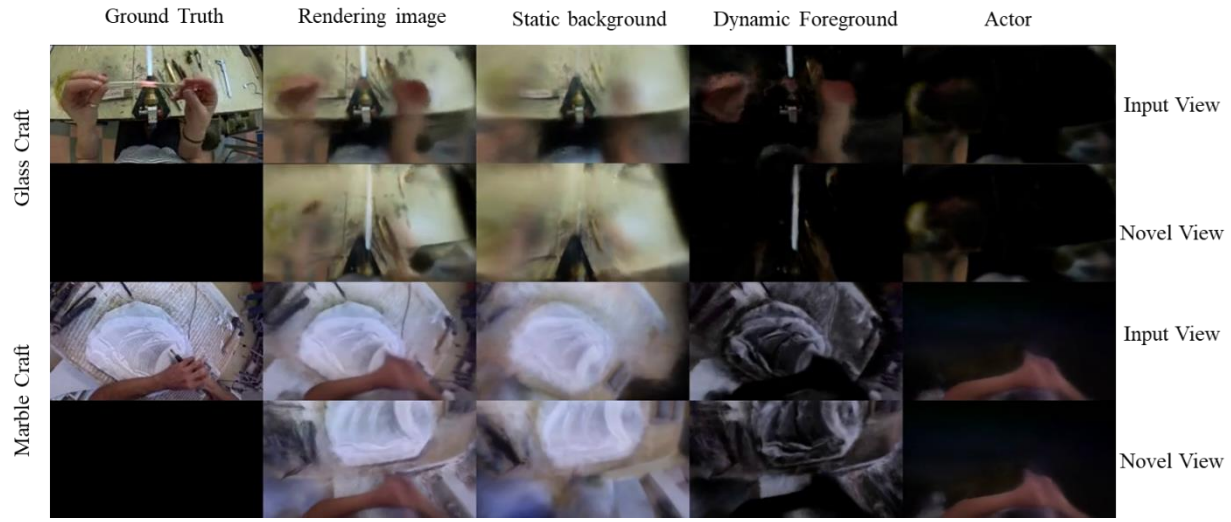

Figure 3:Deformable-3D Gaussian Splatting Visualization Results on glassblowing (top) and marble carving (bottom)

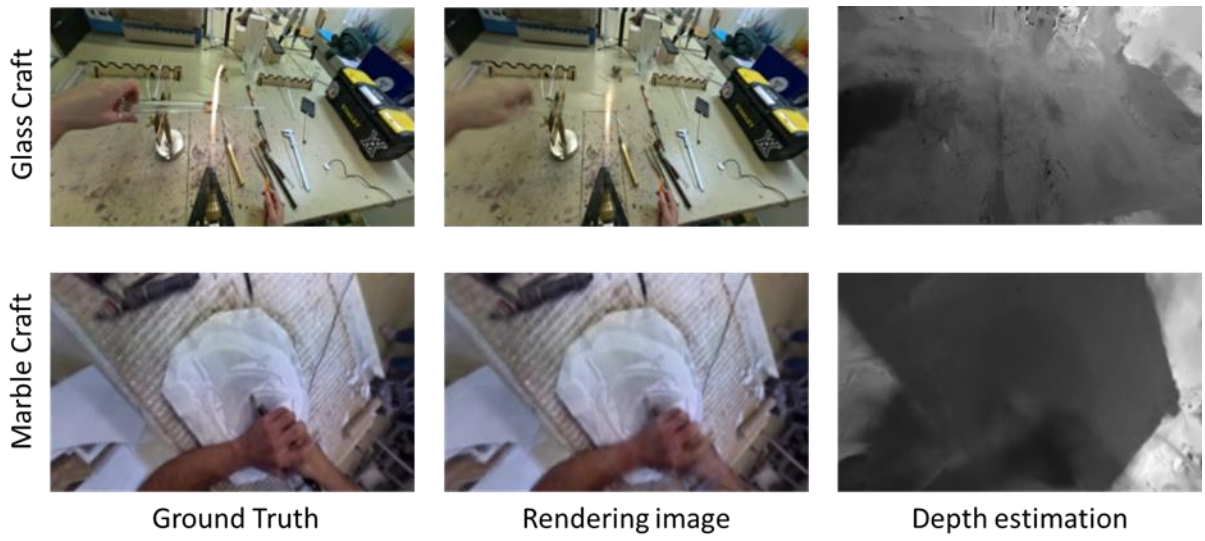

Figure 4. Output generated by the NeuralDiff model for both the marble carving and glass-blowing scenarios

Figure 4 illustrates the output generated by the NeuralDiff model for both the marble carving and glass-blowing scenarios. NeuralDiff employs three parallel MLP layers to segment the scene into three components: static background, dynamic foreground, and the actor (craftsman). Each component is rendered separately and later combined to produce the final image, as shown in the first row for each

case. Furthermore, NeuralDiff allows for novel view generation by fixing a camera perspective, enabling the observation of dynamic scenes from different angles. This novel view rendering is presented in the second row for each scenario. The separation of scene components in NeuralDiff aids in capturing distinct features, but it also necessitates combining these parts to achieve the final rendered output.

In contrast, Figure 5 showcases the output of the Deformable-3DGS model. Unlike NeuralDiff, which separates the scene into different layers, Deformable-3DGS utilizes sparse point clouds derived from the SfM method to represent the entire scene. This approach eliminates the need to separate the scene into static and dynamic objects, allowing the model to generate a novel view in a single rendering pass. Additionally, Deformable-3DGS can output estimated depth maps, providing further insights into the spatial structure of the scene.

The comparison between NeuralDiff and Deformable-3DGS highlights key differences in how these models handle complex dynamic scenes. NeuralDiff's layer-based approach allows for detailed feature capture but requires careful integration of scene components. On the other hand, Deformable-3DGS offers a more unified and efficient rendering process by leveraging point clouds, making it potentially more suitable for real-time applications.

In the next section, we will proceed to the second part of the craft scene, which concerns the activity monitoring, thus the analysis and recognition of the craft gestures.

4. Activity monitoring

As mentioned in the introduction, human movement recognition presents significant challenges due to the complexity of motion data, the need for model generalization across diverse conditions, and the accurate interpretation of dynamic actions. Emerging methods like Meta-Learning (MetaL) and Multi-Task Learning (MultiTL) offer promising solutions by leveraging knowledge exchange and adaptation to new tasks. This work explores the application of these methods to address challenges in human movement recognition, focusing on their potential to improve precision and flexibility in real-time applications. Before moving on to the presentation of methodologies and results, it is important to define the different terms that concern human movement, and further understand its complexity, specifically in crafts.

4.1 Understanding the complexity of the human movement

Many scientific works related to human movement have been using terms such as human gesture, motion, activity, action, and movement in an interchangeable way, which does not always facilitate the process of movement analysis and understanding [4]. For the decomposition of human motion and the training of ML algorithms using motion's low-level representations, it is essential to define the above notions. This decomposition can facilitate the creation of a common ground in which action variabilities and movement patterns will be handled.

More specifically, following a top-down relationship schema, motion is defined as any change of position or location of an object in a specific period [5]. Following, movement is defined as the act or process of change of position or posture of an articulated object, shortly referring to physical actions. Succeeding comes to the notion of action, meaning the purposeful, goal-directed behaviour or sequence of movements. Activity comes in between movements and actions in the hierarchy, indicating a range of actions carried out in a certain period [5]. Lastly comes gesture, as a form of non-verbal communication for interacting with people or using an object [1]. Of all the terms, motion, movement, action, and activity appear to be more generic, while gesture refers to repetitive patterns, providing information, communicating emotions, or even commands without necessarily using words. Gestures evolve within a finite period, and their recognition does not only depend on static postures, but mostly on a sequence of movements. They are required to be analysed as dynamic processes necessitating the capturing of both temporal and spatial parameters. Apart from this, a variability of gestures among different people is also observed, due to their physical characteristics, as well as their current state. Factors such as fatigue, stress, or musculoskeletal burden can cause important variations in the same gesture of an individual. These variations mean that a gesture can be within a range of spatial expressions and temporal dynamics, that depend on the respective use case, complicating the gesture recognition process.

4.2 Modelling gestures through time and space: movement primitives and personal variations

Gestures evolve, as such their recognition does not only depend on static postures, but mostly on a sequence of movements. Thus, gestures are required to be analysed as dynamic processes necessitating the capturing of both temporal and spatial parameters.

Apart from this, a variability of gestures among different people is also observed, due to their physical characteristics, as well as the current state of the person performing a gesture. Factors such as fatigue, stress, or musculoskeletal burden can cause important variations in the same gesture, even interpersonal ones. As such, these variations can mean that a gesture can be within a range of spatial expressions and temporal dynamics, that depend on the respective use-case, complicating the gesture recognition process. Due to these complexities, artificial intelligence modules need to have the ability to adapt to these spatiotemporal variations.

According to the above analysis, a gesture can be defined as a series of movement primitives, P. Movement primitives are the fundamental building blocks that facilitate the comprehension and representation of intricate motion patterns in human movement [13].

As such, a gesture G can be represented through primitives as:

$$G = \{P_1, P_2, P_3, \ldots, P_v\}$$

where $v \in \mathbb{N}$. Each primitive P is characterized by one spatial value S and one temporal value t. Primitives are inherently multi-dimensional, involving movements across the three spatial dimensions x, y, z.

It is important to mention that the primitives that synthesize a gesture are not only sequential but also in a strict order, with a level of variability for each primitive, among similar gestures, that are connected to factors such as fatigue, stress, and level of expertise.

This variability is dependent on the respective use case and the dexterity of the people performing the gestures.

4.3 The proposal of a human movement hierarchy

The use of the tools, the interaction with the materials, as well as the patterns observed in the captured data led to the proposal of a hierarchical scheme of decomposition of professional human movements, shown in Figure 6. The first level of the hierarchy consists of small motion units, called movement primitives, they are non-profession specific and are nine in total (operational primitives), belonging to either the category of the materials, the work objects, the used tools, or the preparation/planning. The second and the third level – actions and activities, respectively – are both profession-specific and are not limited to a specific amount. Each part of these two levels consists of one or more parts of the previous hierarchical level. This is a proposal for operational workspace-based primitives that do not only consider the operator's postures but also their interaction with the tools and materials used to reach the final goal of the work routine. In this work, the proposed movement primitives will be compared to the ones defined by [11], consisting of 28 postures with varying ergonomic risk levels based on the European Assembly Worksheet (EAWS)[1], with the objective of effective work planning (biomechanical primitives). The contribution of the two movement primitives will be examined within the framework of a movement hierarchy.

[1] https://www.ergonomiesite.be/documenten/risicoanalyse/EAWS-v1.3.6.pdf

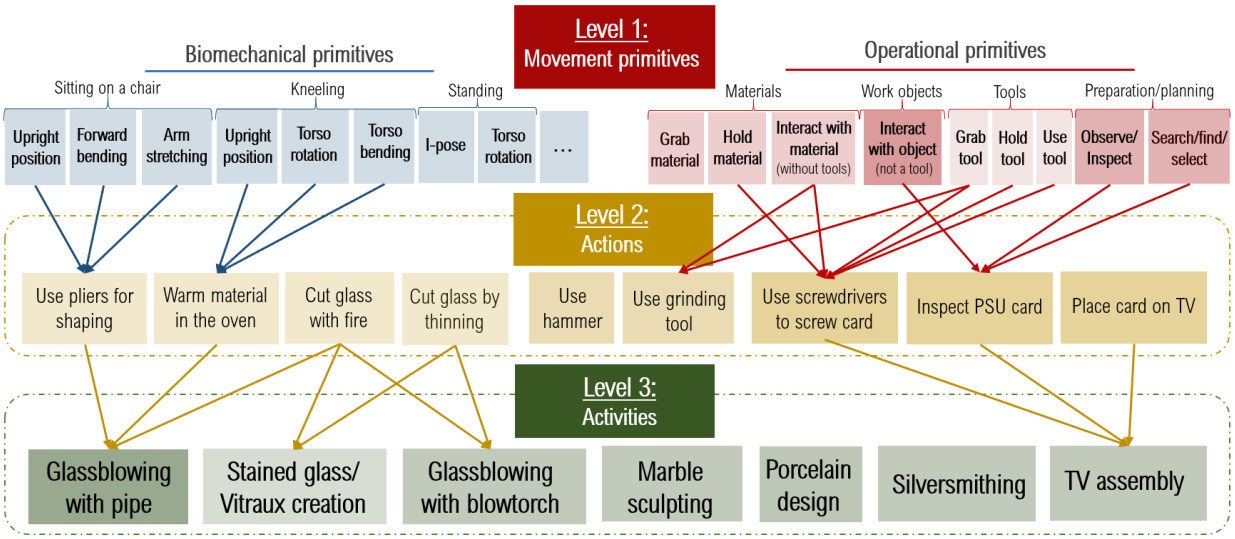

Figure 5: A proposal of a hierarchical scheme of professional human movement.

The main difference between the two primitive proposals comes from the fact that each action in level 2 can be explicitly described by one or more operational primitives, which is not the case for the biomechanical ones. Apart from this, it is important to mention that the biomechanical primitives were captured in a controlled, simulated environment [12], as such they concern non-situated human movements, while the operational primitives are situated ones, as they were captured in real workspaces and concern the implementation of technical knowledge for a specific goal.

A proposed methodology to be followed is shown in Figure 7 and is structured through some first hypotheses:

1. Could the hierarchical structure of situated human motion be considered in MultiTL?
2. Are situated motion primitives more beneficial in recognition accuracy than non-situated ones?
3. Could the knowledge sharing from lower to higher hierarchy levels and its impact on the final recognition accuracy be estimated?

Before delving into the proposed methodology and the implemented experiments, a brief introduction of both MultiTL and MetaL will be presented.

4.4 Meta-Learning and multitask learning

The prefix "meta" is derived from the Greek word μέτα (meta), which can be translated as "after" or "with". In a variety of contexts, the prefix "meta" denotes a higher level of abstraction or something self-referential or self-reflective. MetaL allows a model to learn how to learn by benefiting from prior knowledge to learn a new task, rather than accumulate subject knowledge, and it also appears useful for resolving few-shot learning problems. More specifically, having a set of datasets with loss functions L, The objective is to find an initialization or set of parameters that allows the model to learn and efficiently generalize when new, unseen data appear. Thus, the goal is to optimize these parameters and minimize the loss.

Due to the above definition, the distinction of MetaL from other methods, such as MultiTL, or transfer learning might not be evident. However, the goal of MultiTL is to improve the performance of multiple related tasks in parallel, by leveraging useful information among them, while MetaL aims to find parameters that can enable the respective model to learn new, unseen tasks effectively.

According to the two definitions, MetaL could be combined with MultiTL to produce models with good performance that can adapt to unseen tasks.

On the other side, Transfer Learning refers to a pre-trained model with a source task being reused in a similar task, called the target task. There is no essential goal difference between Meta and Transfer Learning, however, the optimization algorithm is where they diverge the most from one another. While determining priors through learning the source task, Transfer Learning does not have a meta-objective. On the other hand, priors are assessed during the learning of a new task in MetaL, where they are extracted from the outer optimization. In contrast to MetaL, which transfers a range of meta representations, Transfer Learning just shares model parameters.

Multi-task learning (MultiTL) is a sub-field of Machine Learning that seeks to enhance the performance of multiple related learning tasks by exploiting relevant information shared among them. Such a technique presents benefits such as enhanced data efficiency, reduced overfitting via shared representations, and faster learning by leveraging auxiliary data. The goal of MultiTL in this case is the handling of the main challenges in DL, as presented in the introduction, concerning the requirement for a vast amount of data and the corresponding computational power. In the standard techniques of machine learning, there is always a source task and a target task, used along their own, separate model. In real-life scenarios though, dependencies appear and different tasks can have common characteristics and a common structure.

As such, MultiTL leverages this limitation of traditional machine learning, by exploring the inherent task relationships to improve the performance of tasks, by synchronously learning them.

Furthermore, in many cases for standard machine learning methods, the amount of data required for the proper training of the respective algorithm is extraordinary, however, in many applications, such as the medical ones, it is not possible to have access to such a vast amount of data. MultiTL leverages the above problem by exploiting useful information from related tasks. The notion of the "task" in MultiTL is handled by the scientific community in three different ways. As such, different tasks can either denote 1) the classes of a dataset, thus each human action included in an activity, 2) different modalities or types of signal (sound, image, RGB, depth, optical flow), or 3) different processes (i.e. gesture recognition and image segmentation). The objective of MultiTL learning is directly related to the computation of the loss L, derived as a combination of the loss of each one of the tasks in the MultiTL scenario, weighted by the coefficients or weights, defined according to the saliency of each task. This loss aims to improve the performance of the tasks, through backpropagation.

Focusing on human movement, and considering that by default one of the MultiTL tasks will concern human movement recognition, the other tasks, implemented in parallel, can focus either on movement recognition or analysis, or even seemingly different tasks, which can contribute to improving the movement recognition performance.

4.5 Methodology proposal

In an attempt to start addressing the above questions, a first experiment was designed and was based on the work of Olivas et al. [11] that deployed the aforementioned biomechanical primitives to perform dexterity analysis in professional environments. This first experiment concerns the identification of the contribution of those primitives to professional gesture recognition. The second step of this experiment concerns the comparison of the above contribution to that of the proposed operational primitives.

Due to the scientific challenges that concern gesture recognition, thus the ability of AI algorithms to efficiently transfer knowledge and generalize, in the first experiment, an MTL module is used to evaluate the contribution of the biomechanical primitives in gesture recognition. Seven different datasets are used for the specific experiment. The first dataset, which is used as input in Task 1 (movement primitives' recognition), consists of the 28 postures that are based on EAWS, with a few of them being shown in Figure 6. The input in Task 2 (action recognition) consists of 56 actions from 6 manual professional scenarios, including both data from the CRAEFT project (glass blowing), as well as data captured during the extension of this project, the Mingei one (Silk weaving, Glassblowing, and Mastic cultivation). Task 3 (activity recognition) consists of the full routine of the professions mentioned in Task 2, however without them being segmented into smaller units. The data that were used for these 3 Tasks are included in the "Human Motion Capture Benchmark", created by our team[2].

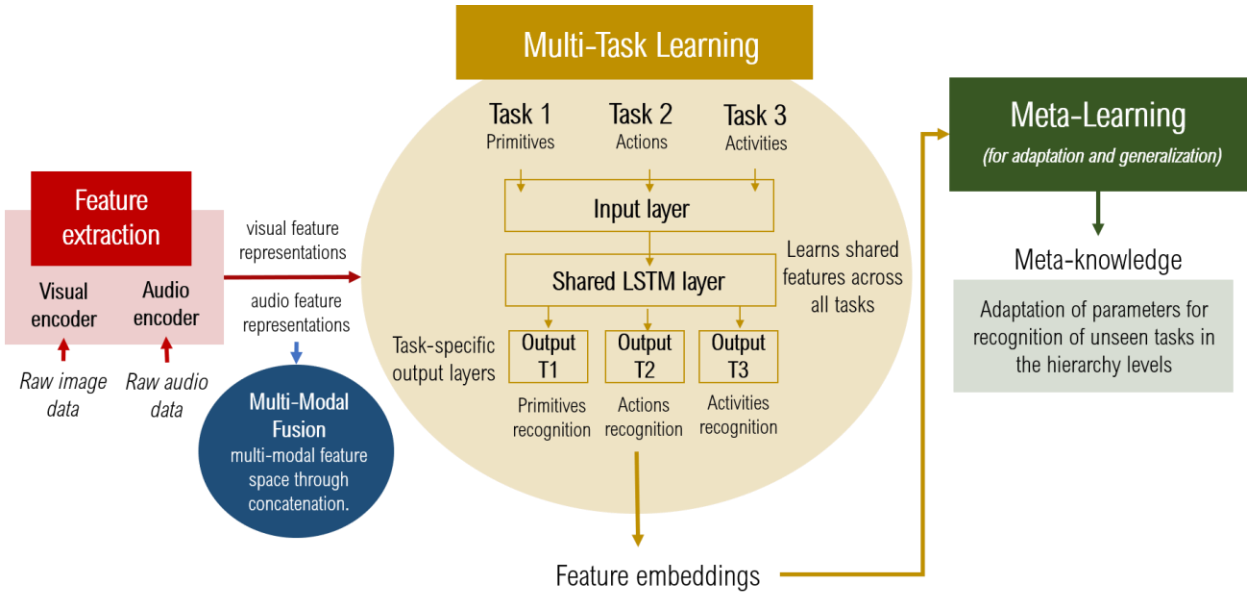

Figure 6: The proposed methodology to be tested. A hybrid version of Multi-Task and Meta Learning could result to a robust model able to extrapolate and generalize.

Concerning the characteristics of these datasets, the ones in Task 1 include 30 repetitions for each one of

[2] https://www.caor.minesparis.psl.eu/human-motion-capture-benchmark/

the 28 postures, from 10 different subjects, which is not the case for the data used in Task 2 and 3, as each repetition was captured by either one or two professional operators.

More specifically, the MTL module that is implemented consists of one input layer, where the three tasks are given as input, one shared LSTM layer, that learns shared features across all tasks, and three task-specific output layers that provide the recognition result for the respective tasks.

More specifically on the shared LSTM layer, it is where knowledge exchange occurs, as it learns representations that are useful for all tasks. The intuition behind multi-task learning is that by learning to perform several tasks at the same time, the model can learn a more generalized representation of the input data. Then, in the task-specific layers, the model learns features specific to each task, based on the representation provided by the shared layers. During the training process, the model receives gradients from the losses of all three tasks, which are back-propagated through it. The shared layers receive gradients from all tasks, thus leading the model to learn representations useful for all of them. Thus, the network is updated to minimize the combined loss function, which according to the standard practices is a weighted sum of the individual sum losses.

However, the task-specific layers receive gradients only from their respective task, making them able to better focus on their task. This structure leads to a way of knowledge exchange where the shared layers perform as a bottleneck that forces the model to find common or commonly important information among the tasks, improving the overall performance.

Table 4: Comparison of the accuracy and loss among the MTL setup and the accuracy and the loss extracted for individually recognizing the tasks in a standard LSTM setup.

Training type	Task	Accuracy (%)	Loss (%)
Single task	1	91	27
MTL	1	93	20
Single task	2	88	34
MTL	2	98.9	4
Single task	3	99	1
MTL	3	100	0.014

Table 5: Ablation study for the contribution of task 1 (T1), task 2 (T2) and task 3 (T3) in the MTL setup.

	T1 accuracy	T2 accuracy	T3 accuracy
MTL (T1, T2, T3)	91%	97%	100%
MTL (T1, T2)	90%	98%	-
MTL (T1, T3)	91%	-	100%
MTL (T2, T3)	-	87%	100%

More specifically, the results of this experiment (Table 4) are presented along with an ablation study that aims to explain the contribution of the three tasks in the specific MTL setup in Table 5. More specifically in Table 4, it is noticed that the recognition accuracy of all three tasks is improved during MTL, in comparison to them being trained individually, with task 2 having the biggest improvement in recognition accuracy +10.9%, with the loss also being minimized by 30% which is quite an important amount.

Concerning Task 1 and Task 2, the recognition accuracy is improved by only +2% for Task 1 and by +1% for Task 2, however, the respective losses are improved by -7% for Task 1 and by almost -1% for Task 3. This observation leads to the conclusion that MTL indeed benefits the recognition of the different hierarchy levels through the knowledge exchange performed, with the actions level being positively affected the most.

The reduction of the loss across all the tasks in the MTL setup indicates that the MTL model is more efficient in minimizing errors. The significant reduction in loss for Task 1 and 2 might indicate that these tasks benefit from shared learning dynamics, which is normal due to the characteristics of the three tasks of the hierarchy levels.

To further examine the contribution of each one of the tasks within MTL, an ablation study where each one of the tasks was removed from the input layer was performed. The results are shown in Table 5 and show a significant reduction in the accuracy by 10% when Task 1 is removed from the MTL setup. This indicates that movement primitives positively lead to a better understanding and recognition of human actions in professional environments. In the same table, it is shown that when the actions or the activities are removed, the accuracy of the other task is not affected, which leads to a first observation that movement primitives are the stepping stone for human movement hierarchy and that further experiments are required to identify the amount of the required hierarchy levels.

The aforementioned experiments indicate an important contribution of both the use of a hierarchical scheme in gesture recognition, as well as of movement primitives in MTL and are further examined to identify the exact contribution of each hierarchical level to the other in MTL, through the backpropagated loss and weights to the networks. The main goal here is to identify if they indeed take into consideration the characteristics of situated gestures or if in the specific case, the calculation of the backpropagated loss needs to be modified. More specifically since the proposed hierarchy is thought to provide spatial and temporal information, as well as information about the dexterity of the gesture and the used tools and materials, it will be studied whether these details are indeed transferred through the shared gradients or not and an effort will be done to modify the shared information accordingly.

As such, a first test on the way that the current loss is calculated in MTL is performed. The aim here is to implement an MTL algorithm that will be able to capture and learn all the important characteristics of all Tasks and crafts it has been trained with, optimally, to then move forward to a Meta-Learning approach, where the implemented module will be able to not only be confident with specific crafts but will also be able to recognize new unseen ones. This will decrease the needed time for the creation of new training installations, and 3D simulations.

In the following experiment, we propose a new approach to calculating the loss in our MTL framework, that includes a dynamic knowledge exchange between the different levels of hierarchy, to identify if this exchange needs to be adapted while training, to further optimize the algorithm's performance. Loss functions are critical in training machine learning models, as they quantify the difference between the predicted output and the ground truth, guiding the model's learning process. By minimizing this loss, the model iteratively improves its predictions, becoming more accurate over time. In the context of MultiTL, an optimized loss function is even more crucial because it determines how effectively the shared and task-specific layers learn from the combined tasks. A well-designed loss function can enhance the model's ability to capture the nuances of each task, leading to better generalization and transferability across

different tasks and datasets. Our goal is to develop a loss calculation method that not only minimizes error but also ensures that the model learns the most relevant features for each task, particularly in recognizing complex human movements across various professional crafts. By refining how loss is calculated and optimized, we aim to create an MTL algorithm that can more effectively integrate and learn from the diverse tasks it is trained on, ultimately improving the model's overall robustness and accuracy.

The idea of dynamic knowledge exchange refers to the model's ability to transfer information between the shared and task-specific layers during training in a flexible manner. Rather than treating the knowledge flow as static (i.e., fixed during the whole training process), this approach suggests that the exchange of information should adapt based on the evolving needs of each task. This dynamic adjustment can involve altering the contribution of different layers to the overall learning process depending on the current state of training or the specific nuances of each task. For example, as the model learns, it might discover that some layers or features are more beneficial for certain tasks and adjust the flow of information accordingly.

This enhancement is expected to pave the way for the algorithm's future application in a MetaL context, where it will need to adapt to new, unseen crafts and gestures with minimal additional training.

The standard way that loss is calculated in MultiTL, is through a weighted sum of the individual task losses, thus:

$$L_{total} = w_1 L_1 + w_2 L_2 + w_3 L_3$$

where L_{total} is the total loss for all tasks, L_1, L_2, L_3 are the individual task losses, and w_1, w_2, w_3 are weights, thus coefficients that compute the significance of each task, and the individual loss of each one of the tasks is computed through the categorical cross-entropy loss function:

$$L_1 = \frac{1}{N} \sum_{i=1}^{N} \sum_{C=1}^{C} y_i \, log(\hat{y}_i) \qquad (1)$$

were N, is the number of samples provided for training, C the number of classes for each task, y_i is the input label and $\hat{y}_i$ the predicted label.

Even though a few works have focused on the way that the loss is computed in MultiTL settings, to improve generalization, the final results do not justify their goal.

In this specific case, it was noticed that through the standard approach of the loss computation, for each epoch in the LSTM network, the individual losses are computed through equation (1), without encapsulating the loss of the previous epoch in them.

As such, we proposed a new way of calculating the total loss, that follows the format of the observation equation in a state space representation, and is as follows:

$$L_{total_{epoch}} = \alpha_1 Loss_{total_{epoch-2}} + \alpha_2 Loss_{total_{epoch-1}} + w_1 L_1 + w_2 L_2 + w_3 L_3$$

(2)

Where the total loss is affected not only by the weighted individual task losses but also by the weighted total losses of the two previous epochs in the LSTM. The coefficients $\alpha_1, \alpha_2, w_1, w_2, w_3$ are calculated through the process of Maximum Likelihood Estimation. This way the total loss explicitly encapsulates the memory of the previous errors in its computation. The extracted total loss in each epoch is backpropagated to the network to help it gradually improve. The weights assigned to the previous epoch losses determine the extent to which past learning impacts current model adjustments. These weights can be interpreted as indicators of knowledge retention**,** where higher weights suggest a greater reliance on the knowledge acquired in previous epochs. This retention reflects the model's ability to carry forward valuable information from past learning cycles, effectively influencing the current training phase. Rather than directly exchanging knowledge between tasks at a single point in time, this process captures a temporal knowledge retention mechanism, where the model leverages the accumulated experience to refine its understanding and performance in subsequent epochs. By incorporating this temporal aspect, the model can maintain continuity in learning, ensuring that important insights from prior epochs are not lost but are instead integrated into ongoing training efforts, leading to a more robust and well-informed learning trajectory.

Various experiments have been performed, to see which setup provides the optimal results. More specifically, the tests involved using static or dynamic coefficients, backpropagating or not backpropagating the total loss (Table 5).

Table 6: Accuracy results of different experiments performed for the MultiTL approach. The baseline is the result MTL(T1, T2, T3) presented in Table 5.

	T1 accuracy	T2 accuracy	T3 accuracy
Baseline MTL	91%	97%	100%
MTL without coef. updates, 40 epochs, eq2	92%	98.5%	100%
MTL with coef. updates, 40 epochs, eq2	91.8%	98.6%	100%
MTL with coef. updates, 80 epochs, eq2	95%	99.7%	100%
MTL with coef. updates & backprop., 80 epochs, eq2	94%	99.7%	100%

The results of the various MultiTL setups provide valuable insights into the impact of different loss calculation strategies on model accuracy across the three tasks. The baseline MultiTL setup achieved accuracies of 91% for Task 1, 97% for Task 2, and 100% for Task 3. These results indicate a strong initial performance, particularly for Task 3, which reached perfect accuracy. However, there is still room for improvement, especially in Task 1, where the accuracy is somewhat lower compared to the other tasks.

In the experiment where the model was trained for 40 epochs without updating the loss coefficients, there was a slight improvement in accuracy across all tasks. Task 1 improved to 92%, Task 2 to 98.5%, and Task 3 maintained its perfect accuracy. This suggests that even without coefficient updates, additional training epochs help the model to refine its understanding and improve performance slightly. However, the improvements were relatively modest, indicating that more sophisticated adjustments might be necessary to achieve significant gains.

When coefficient updates were introduced while training for 40 epochs, the accuracy of Task 2 increased marginally to 98.6%, while Task 1 saw a slight decrease to 91.8%, and Task 3 remained at 100%. The slight decrease in Task 1 accuracy suggests that the updates to coefficients may have slightly disrupted the balance in learning for this task, though the impact was minimal. This result indicates that while coefficient updates can be beneficial, they need to be applied carefully to avoid unintended effects on certain tasks.

Extending the training to 80 epochs with coefficient updates yielded significant improvements, particularly for Task 1 and Task 2. Task 1 accuracy increased notably to 95%, and Task 2 almost reached perfect accuracy at 99.7%. Task 3 continued to maintain 100% accuracy. This indicates that extended training with adaptive coefficients helps the model to better optimize the learning process, especially for tasks that initially had lower accuracy. The extended training period allowed the model to better adjust to the complexities of each task, leading to more robust and accurate performance.

Finally, with both coefficient updates and backpropagation of the total loss across 80 epochs, Task 1 accuracy slightly decreased to 94%, while Task 2 maintained its high accuracy at 99.7% and Task 3 continued to perform perfectly. This result suggests that while the backpropagation of the total loss across epochs helps maintain high performance, it may introduce a small amount of variability in the model's ability to generalize across all tasks, particularly for Task 1. The slight decrease in Task 1 accuracy might indicate a trade-off where the additional information from backpropagating the total loss provides benefits overall but also introduces complexity that can affect certain tasks differently.

Overall, the experiments show that coefficient updates, as well as generally contribute to improved accuracy, especially when combined with extended training. This is particularly evident in the improvements seen in Task 1 and Task 2 when training was extended to 80 epochs. The results highlight the importance of sufficient training duration in MTL settings to allow the model to fully learn and adapt to the complexities of multiple tasks. Additionally, the incorporation of backpropagation of the total loss, while beneficial for overall model performance, introduced a slight decrease in accuracy for Task 1. This suggests a potential trade-off where backpropagating the total loss helps in fine-tuning the model but might also introduce some instability in learning across tasks.

The results indicate that the proposed adjustments to loss calculation and the introduction of coefficient updates, along with extended training, significantly enhance the performance of the MTL model, particularly for tasks that initially had lower accuracy. However, the slight variability introduced by backpropagating the total loss suggests the need for further refinement in how these adjustments are implemented to ensure consistent improvements across all tasks. Task 3 consistently achieved 100% accuracy across all setups, indicating that it might be an easier task for the model to learn or that the task's characteristics align well with the MTL framework. In contrast, Task 1 showed more variability, highlighting the need for tailored strategies in MTL to balance learning across tasks with different complexities.

5. Conclusion and future steps

In conclusion, this work has demonstrated the significant potential of combining advanced machine learning methods, such as Neural Radiance Fields (NeRFs), 3D Gaussian Splatting (3DGS), and Multi-Task Learning (MTL), to enhance the preservation, understanding, and transmission of traditional crafts. By utilizing egocentric video and dynamic 3D scene reconstruction, we were able to capture the intricate details of craft processes, offering a more immersive and detailed view compared to traditional 2D video approaches. This provides valuable insights into the spatial arrangement and complex movements involved in these crafts, which are crucial for both vocational training and the digitization of Intangible Cultural Heritage (ICH).

The application of MTL to the recognition of human movements within these crafts revealed the benefits of hierarchical modelling of tasks, where movement primitives, actions, and activities were effectively recognized with improved accuracy. The experiments highlighted the importance of optimizing loss functions within the MTL framework to better capture the nuances of each task and improve the model's generalization across different datasets and conditions. By introducing a new approach to loss calculation that accounts for the memory of previous errors, we were able to enhance the robustness and accuracy of the model, particularly for complex and dynamic tasks.

Moreover, the results of this work underscore the value of extending training duration and incorporating coefficient updates to achieve significant gains in performance, especially for tasks that initially posed more challenges. The integration of backpropagated total loss also contributed to the model's overall efficiency, though it introduced some variability that warrants further exploration and refinement.

Overall, the methodologies and findings presented here contribute to the ongoing effort to develop more sophisticated and adaptable machine-learning tools for the preservation of traditional crafts. The combination of scene understanding and activity monitoring through advanced ML techniques paves the way for more precise, flexible, and real-time applications in cultural heritage preservation and vocational training. Future work will focus on further refining these models, particularly in the context of Meta-Learning, to ensure they can adapt to new, unseen crafts and gestures with minimal retraining, thereby reducing the time and resources required for new training installations and simulations. On what concerns scene understanding, NeuralDiff and Deformable-3DGS provide satisfactory results, however, they lack detail, when representing instant actions, which are crucial for understanding hand motions, even though they have proven to provide a good performance in dynamic scene representation. Instant motions are essential for comprehending hand motions as they involve changes in motion when a tool is switched. Therefore, the next step is to focus on further studying instant motions, such as hand movements and interactions with tools.

From the current perspective, the small number of frames led to low-quality instant action representation. Additionally, the scenes are not well separated, making it difficult to represent both dynamic and static scenes together, which increases computational dependence. To address this, we propose introducing masks to distinguish between the dynamic and static parts (i.e. a static workbench and the dynamic

movement of the tools and the hands of the human operator. This will enable the hierarchical rendering, as well as the combination of both the static and the dynamic elements.

References

1. Manitsaris, Sotiris, Gavriela Senteri, Dimitrios Makrygiannis, and Alina Glushkova. "Human movement representation on multivariate time series for recognition of professional gestures and forecasting their trajectories." *Frontiers in Robotics and AI* 7 (2020): 80.
2. Morshed, Md Golam, Tangina Sultana, Aftab Alam, and Young-Koo Lee. "Human action recognition: A taxonomy-based survey, updates, and opportunities." Sensors 23, no. 4 (2023): 2182.
3. Jegham, Imen, Anouar Ben Khalifa, Ihsen Alouani, and Mohamed Ali Mahjoub. "Vision-based human action recognition: An overview and real-world challenges." *Forensic Science International: Digital Investigation* 32 (2020): 200901.
4. Moeslund, Thomas B., Adrian Hilton, and Volker Krüger. "A survey of advances in vision-based human motion capture and analysis." *Computer vision and image understanding* 104, no. 2-3 (2006): 90-126.
5. Schambra, Heidi M., Avinash Parnandi, Natasha G. Pandit, Jasim Uddin, Audre Wirtanen, and Dawn M. Nilsen. "A taxonomy of functional upper extremity motion." *Frontiers in Neurology* 10 (2019): 857.
6. Mescheder, Lars, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. "Occupancy networks: Learning 3d reconstruction in function space." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4460-4470. 2019.
7. Mildenhall, Ben, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. "Nerf: Representing scenes as neural radiance fields for view synthesis." *Communications of the ACM* 65, no. 1 (2021): 99-106.
8. Kerbl, Bernhard, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. "3D Gaussian Splatting for Real-Time Radiance Field Rendering." *ACM Trans. Graph.* 42, no. 4 (2023): 139-1.
9. Yang, Ziyi, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. "Deformable 3d Gaussians for high-fidelity monocular dynamic scene reconstruction." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20331-20341. 2024.
10. Tschernezki, Vadim, Diane Larlus, and Andrea Vedaldi. "NeuralDiff: Segmenting 3D objects that move in egocentric videos." In *2021 International Conference on 3D Vision (3DV)*, pp. 910-919. IEEE, 2021.
11. Olivas-Padilla, Brenda Elizabeth, Alina Glushkova, and Sotiris Manitsaris. "Motion capture benchmark of real industrial tasks and traditional crafts for human movement analysis." *IEEE Access* 11 (2023): 40075-40092.
12. Olivas-Padilla, Brenda Elizabeth, Dimitris Papanagiotou, Gavriela Senteri, Sotiris Manitsaris, and Alina Glushkova. "Improving Human-Robot Collaboration in TV Assembly through Computational Ergonomics: Effective Task Delegation and Robot Adaptation." In *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 480-487. IEEE, 2023.
13. Arbib, Michael A. "Perceptual structures and distributed motor control." *Comprehensive physiology* (2011): 1449-1480.