A TRANSITION TOWARDS VIRTUAL REPRESENTATIONS OF VISUAL SCENES

Américo Pereira[1,2], Pedro Carvalho[1,3] and Luís Côrte-Real[1,2]

[1]Centre for Telecommunications and Multimedia, INESC TEC, Porto, Portugal
[2]Faculty of Engineering, University of Porto, Porto, Portugal
[3]Polytechnic of Porto, School of Engineering, Porto, Portugal

ABSTRACT

We propose a unified architecture for visual scene understanding, aimed at overcoming the limitations of traditional, fragmented approaches in computer vision. Our work focuses on creating a system that accurately and coherently interprets visual scenes, with the ultimate goal to provide a 3D virtual representation, which is particularly useful for applications in virtual and augmented reality. By integrating various visual and semantic processing tasks into a single, adaptable framework, our architecture simplifies the design process, ensuring a seamless and consistent scene interpretation. This is particularly important in complex systems that rely on 3D synthesis, as the need for precise and semantically coherent scene descriptions keeps on growing. Our unified approach addresses these challenges, offering a flexible and efficient solution. We demonstrate the practical effectiveness of our architecture through a proof-of-concept system and explore its potential in various application domains, proving its value in advancing the field of computer vision.

KEYWORDS

Visual Scene Understanding, Scene Understanding, 3D Reconstruction, Semantic Compression

1. INTRODUCTION

Visual scene understanding is a fundamental task in computer vision that aims to extract rich and meaningful information from visual data. It plays a crucial role in numerous real-world applications where perception and interpretation of visual information is required to assess and complete different tasks.

Nowadays, with the increased attention towards virtual reality, augmented reality and overall interest in providing richer forms to visualize data, it becomes clear that there is a need to integrate 3D techniques and methods with visual scene understanding. Hence, the task of automatic visual scene understanding for 3D scene synthesis can be seen as a new challenge. This involves automatic perception, analysis and interpretation of visual data that can be employed into a dynamic 3D scene through the usage of multiple sensors and algorithms. This new challenge can see application in multiple application scenarios, such as: surveillance, sports, retail or entertainment. As an example, in [1] visual data and synthesis are used to create a mixed reality system that allows users to explore a 3D environment.

David C. Wyld et al. (Eds): IoTE, CNDC, DSA, AIAA, NLPTA, DPPR - 2024
pp. 119-137, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.142310

Traditional approaches to scene understanding often involve separate and specialized algorithms for different tasks, leading to fragmented and disjointed analysis that hinders the system's ability to achieve a holistic and coherent understanding of visual scenes. As observed in [2], there is evidence of a need for a well-structured and unified framework that is capable of analysing a scene, describing and synthesizing it. This could provide several advantages over traditional disjointed approaches, such as allowing for a seamless integration of different modules, facilitating information exchange and enabling synergy among tasks. Visual scenes are composed of diverse objects, spatial relationships, contextual cues, and temporal dynamics, which collectively contribute to the overall understanding; thus, it becomes necessary to formulate a cohesive framework that enables a comprehensive and contextually aware understanding of the visual scene by leveraging different types of information. In such a framework it is important that the knowledge extracted from the visual scene is accessible through the entire processing chain, as multiple algorithms that are part of the framework may use this information to enhance the overall understanding of the scene. Contextual understanding is another crucial aspect that a unified framework can address, as scenes are not merely a collection of objects but are characterized by spatial layout, temporal dynamics and semantic coherence.

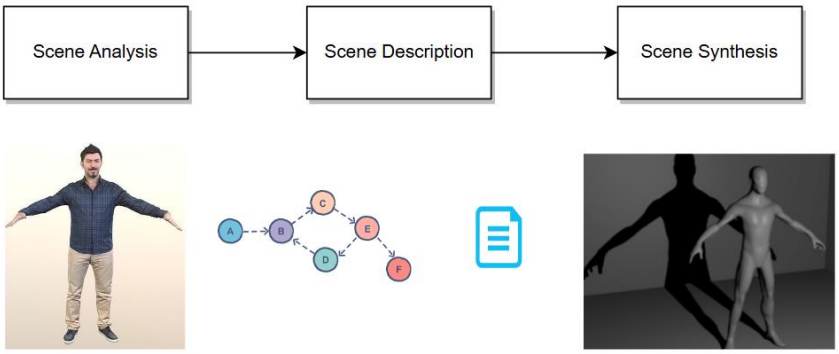

Figure 1. Visual-Virtual translation pipeline initially proposed in [2].

In this article we propose an architecture that enables the creation of a unified framework or system that addresses the challenges of visual scene understanding for 3D scene synthesis. To achieve this, we start by leveraging the initial basic architecture proposed in [2], depicted in Figure 1, and expand the modules, detailing aspects of the framework. Our proposal consists of four main components: scene analysis, scene description, scene synthesis and a data orchestrator. In the scene analysis module, visual input is processed to extract low and high-level features, detect objects, infer semantic segmentation, estimate poses, and capture contextual relationships among objects. This information serves as the foundation for subsequent stages, facilitating a detailed understanding of the scene. The scene description module takes the output of the scene analysis and constructs a high-level representation of the scene that incorporates the spatial information, object attributes, semantic labels and contextual information to generate a structured scene description. Finally, the scene synthesis module utilizes the scene description to generate a realistic and immersive 3D representation of the scene. This generation can blend semantic data, spatial arrangement, and time-based elements to create realistic scenes for specific purposes, offering adaptable and flexible solutions based on input restrictions and output needs. The data orchestrator is responsible to ensure a common ground on all of the processes and concepts within the system, effectively guaranteeing consistency of the data flow across the entire architecture. It also includes an important sub-module that helps in the creation of an informed decisions on the best algorithm combinations to be applied to the input data.

The contributions of this work are three-fold: 1) we showcase the new challenge of visual scene understanding for 3D scene synthesis; 2) we present an unified and flexible system architecture to take on this challenge; and 3) we show a practical application of this architecture by implementing a proof of concept system that incorporates our designs and provide examples of generated hybrid scenes that can be obtained by our system, illustrating the capability of generating synthetic data that could be used to train other models. We also present a series of possible applications that could leverage our proposal to target specific problems.

The document is structured as follows: Section 2 explores existing works in the field of visual scene understanding and discusses existing methodologies, algorithms, and frameworks used in scene analysis and synthesis. Section 3 presents the proposed unified architecture in detail, providing an overview of the architecture as a whole and explaining the role of each component and their interactions within the system, exploring possible technologies and algorithms that can be applied in each component. Section 4 delves into potential use cases and areas of application that benefit from employing our proposal. Section 5 presents a proof-of-concept system that incorporates the main ideas of our proposal into a system and exemplifies possible outcomes that can be obtained. Finally, section 6 summarizes the contributions of the article and discusses future research opportunities and directions required to further improve the proposed system architecture.

2. RELATED WORK

With the improvement of processing power and neural network design, several areas of scene understanding have naturally evolved, with the proposal of new methods and more detailed datasets. When looking at image recognition, works such as NFNets [3] show that it is possible to achieve high accuracy on large image datasets such as ImageNet [4] with a faster training process. RepMLP [5] shows that incorporating prior information into fully connected layers enhances image recognition abilities. Video object segmentation has also evolved, with works such as: SwiftNet [6] that uses pixel-adaptive memory and pixel-wise memory update and match to reduce temporal and spatial redundancy, enabling real time processing; and LCM [7], which also uses a memory-based approach into a semi-supervised method that addresses the problem of not using the sequential order of the frames and object-level knowledge. Another related topic in visual scene understanding is salient object detection; the work presented in [8] studied and compared several approaches, ultimately concluding that there are still many under-explored problems in achieving efficient and reliable network designs. In a recent work, the algorithm IDYOLO [9] is proposed to achieve real-time salient object detection by extending the well-known YOLOv3 [10] algorithm with the instance segmentation algorithm Poly-YOLO [11].

Considering an hierarchical perspective over a scene, detection and tracking can be seen as starting points of a more complete understanding of the information present. Hence, more high-level subjective aspects, such as the meaning of the location of the objects, activities or even the interactions that occur are important and, therefore, a semantic parsing of visual scenes is necessary. A way to amass and convey these details extracted from a visual scene is through the usage of a Scene Graph; which is a data structure that is mainly used to describe objects, attributes and their relationships. It can represent the semantic details of a scene by explicitly modelling objects along with their attributes and relationships. They were originally introduced in Johnson et al. [12] and, since then, research on their generation and application to multiple scenarios has progressed. Scene graphs have been used for tasks such as image/video captioning [13, 14, 15], visual question answering [16, 17, 18], image retrieval [19, 20] or image generation [21, 22]. Despite the research interest in scene graphs, most of the existing works are related to generating the graphs from single images. In the case of videos, there are approaches that use spatio-temporal scene graphs to model the semantic information present in the sequence;

however, due to the constraints that are introduced due to temporal observations, the process of generating the scene graph becomes increasingly difficult. Works such as [23, 24] try to use state-of-the art video object detection and tracking methods to generate the graphs and the results obtained are starting to become more accurate. Figure 2 depicts an example of a scene graph, where objects, attributes and relationships represent the semantic information of the image.

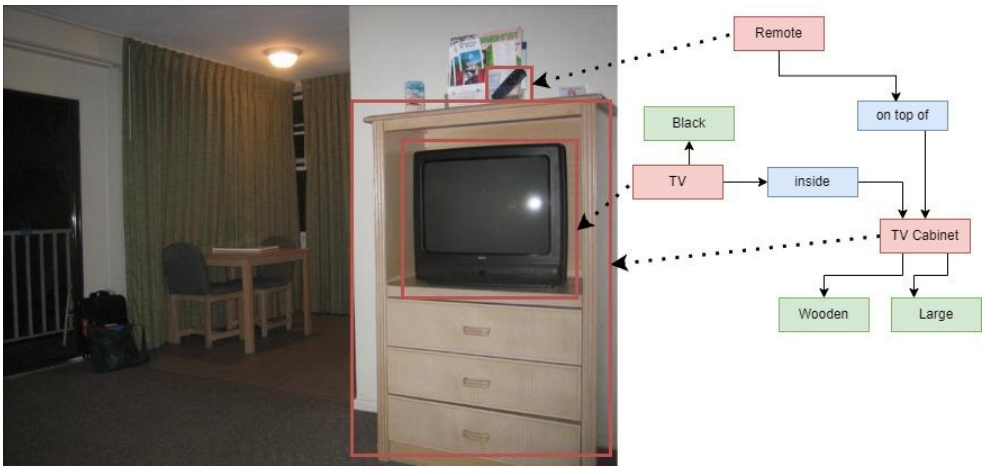

Figure 2. Simple example of a Scene Graph. In red we have objects, in green attributes and in blue relationships.

Human activity detection is also a very challenging and studied topic where the improvements of computational capabilities and neural networks enabled considerable advances. In [26], an RNN with LSTM was used to learn long-term temporal relationships in order to achieve spatio-temporal human action recognition in long videos that have overlapping actions. In a different field, the work presented in [27] detects street-crossing pedestrians for a safer autonomous driving system. Multiple state-of-the-art works are also explored in detail in [28], where action recognition algorithms are compared in multiple application scenarios. Pose estimation is also a powerful tool to assess human activity and in [29] 2D skeleton-based action recognition methods that estimate the pose of humans from RGB images are compared and assessed. Analogous is a study for 3D skeleton-based action recognition [30].

When looking at works that specifically mention visual scene understanding, it is noticeable that it is viewed mostly as a fixed concept. For instance, in [31] RGB and thermal images are used on a multitask-aware network that mixes semantic information with coarse features at various abstraction levels to ultimately segment images. In [32] a deep learning framework for future video prediction is presented, where the authors incorporate a module for scene understanding that serves to reconstruct semantic segmentation and depth images and predict optical flows. A bidirectional projection network is proposed in [33] to leverage the complementary information of 2D and 3D data to provide, once again semantic segmentation, but for 2D and 3D. Semantic scene completion is another related topic, where a 3D scene is reconstructed by leveraging visual and semantic data extracted from single-view depth or RGBD images [34, 35]. There is also work on end-to-end semantic instance reconstruction from incomplete point clouds [36, 37]. These works ultimately show that visual scene understanding has a vast area of application. However, there is a tendency to link the concept towards semantic segmentation and not to the more general idea of extracting semantic data from visual scenes.

As with other areas, 3D virtualization has also evolved in recent years. In particular, human parametric models have been used for multiple scenarios such as 3D human pose and shape

estimation [38], controllable 3D human synthesis [39] or virtual try-ons of clothing [40]. There has also been research on using graph convolutional neural networks to generate 3D human shapes with better resolution [41]. Also, 3D modelling tools and game engines such as Unity [42], Unreal Engine [43] or Blender [44] have also evolved, introducing new features and more support for new graphic interchange languages such as Universal Scene Description (USD) [45], which is a universal format for 3D graphics. However, its usage for 3D reconstruction algorithmic pipelines is still extremely uncommon, showing that its inclusion will lead to new research opportunities. Nowadays, with the usage of Generative Adversarial Networks (GANs) [46, 47, 48] or Neural Radience Fields (NeRF) [49, 50], there has been a significant increase in the realism of the generated images and 3D representations. However, there is still a lack of usability when integrating a less restrictive and versatile application scenario.

3. UNIFIED ARCHITECTURE

Embracing a broader vision of visual scene understanding than what is generally found on literature, we target the paradigm of visual scene understanding for 3D scene synthesis and explore the idea of a unified architecture that targets the processes required to transit from visual to semantic data, and further to a posterior 3D reconstruction. The proposed architecture, depicted in Figure 3, employs four essential modules: scene analysis, scene description, scene synthesis and a data orchestrator. One support module is also present. In essence, the scene analysis module processes visual data, extracting key information like object detection and spatial relationships. The scene description module then constructs a high-level representation of the scene, capturing attributes and contextual details. Finally, the scene synthesis module uses this information enabling the creation of flexible, customizable and realistic 3D scene, incorporating semantic data, spatial layout, and temporal dynamics for an immersive experience. The data orchestrator serves as a central hub for defining and sharing data, such as object types, attributes, and relationships. This way, the standardization of the knowledge domain promotes consistency and interoperability across the system, enabling seamless communication between components. The algorithm selector sub-module is a dynamic component that assesses user input and considers factors like scene type, complexity, resources, and desired output to then intelligently choose which scene analysis algorithms to run and what type of information needs to be included in the scene description.

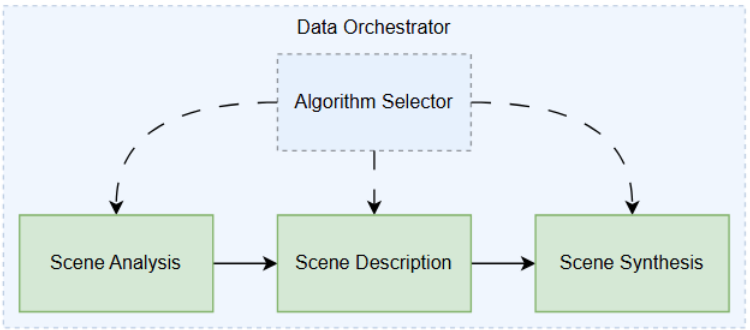

Figure 3. Proposed high level architecture for transitioning from visual scene towards 3D synthesis. The three main modules are depicted as green boxes and the two supporting components that glues the architecture as blue boxes.

By incorporating these components, we allow a system that implements this architecture to be able to dynamically adapt to different scenes and requirements, thus optimizing the performance and output of the system. In the following subsections we delve into each of these components

and detail their internal structure and what type of data flow and algorithms could be used for their implementation.

3.1. Algorithm Selector

The Algorithm Selector intends to provide flexibility to the entire system. It receives user input information and is responsible for interpreting this information and provide a selection of the most appropriate algorithms to be used to process the input video or images. The decision must take into consideration the type of scene that is to be analyzed, the type of information that is to be described and the desired output to effectively select the most appropriate group of algorithms. In the current proposal, the algorithm selector follows a rule-based selection process but can evolve into a more sophisticated process. By making use of rule sets provided in the configuration, the Algorithm Selector is able to provide a deterministic selection of algorithms and associated parameter to process input visual data. This allows for fine-grained control over the algorithms so that they can target specific problems that can appear and are accounted for by the rule sets.

3.2. Data Orchestrator

Visual scene understanding requires the integration of multiple algorithms, addressing different types of information that can be extracted from a scene. This means that for a system to incorporate an ensemble of algorithms, it needs to have a common ground where the type of data and the flow of information is well structured, thus enabling the adaptation of multiple algorithms and enabling future development without requiring structural re-definitions of the entire architecture. This can be critical, for instance, to reduce latency in critical applications such as self-driving cars or adjusting UAVs depending on their surroundings and sensory information. To address these issues, the Data Orchestrator module is proposed, which binds and connects all components in the system. For this module we propose the usage of a knowledge base or ontology that provides a shared and consistent representation of objects, attributes, and relationships, ensuring a common understanding across algorithms. To achieve this, a structured representation of the knowledge base/ontology needs to be designed, so that it captures the semantics and relationships between different concepts in the scene understanding domain.

The structured representation can be implemented using various technologies and standards such as RDF (Resource Description Framework) [51], OWL (Web Ontology Language) [52], or JSON-LD (JSON for Linked Data) [53]. This can then be populated by integrating existing domain-specific knowledge sources, such as existing datasets like MS COCO [54], Open Image [55] or Image Net [4]; or external ontologies [56, 57, 58]. Algorithms within remaining modules can then access and query the knowledge base/ontology to obtain relevant information for their respective tasks.

3.3. Visual Scene Analysis

The Visual Scene Analysis module is the first step in the processing chain of the system. Visual input data in given to the module in the form of images or videos and the algorithms identified by the Algorithm Selector sub-module are applied to the data to extract meaningful information about the scene. As different types of information can be extracted, it employs a multi-stage processing pipeline to perform various tasks and ensure that processes required by multiple algorithms are not performed more than once.

Internally, the module should have a hierarchical structure that goes from basic pre-processing techniques, such as noise reduction, image enhancements, normalization, resizes and other low-

level image manipulation techniques; to more advanced techniques. Naturally, the processes required depend on the input data and on the requirements specified by the user and associated rule sets. These advanced techniques include processes such as: object detection and recognition, semantic segmentation, pose estimation, and so on. Additionally, other more specialized and advanced algorithms may also be integrated, based on the rules specified for the scene analysis algorithms. These may include scene classification, object tracking, action recognition, group behaviour and so on. We can see in Figure 4 an illustration of this module, where the hierarchy of the processing pipeline within the module can be observed.

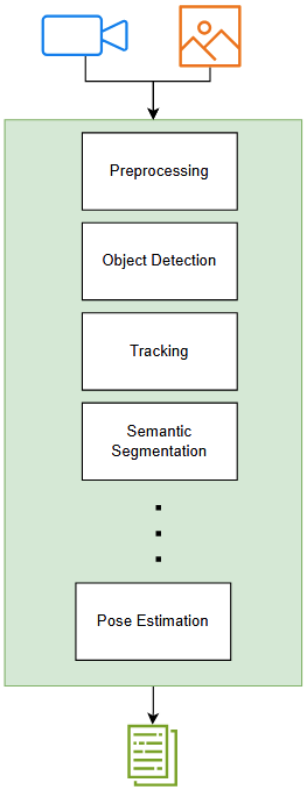

Figure 4. High level scheme of the visual scene analysis module. At the top we can have preprocessing techniques that are followed by more advanced processes to extract meaningful information. These processes may or may not be interconnected and depend on the application scenario and the rules specified as well as the desired output. The output of the module is processed data that is the forwarded to the scene description module.

Overall, the internal architecture of the Visual Scene Analysis module incorporates a combination of traditional computer vision techniques and state-of-the-art deep learning-based methods to extract rich, high level and meaningful information from the input data. The combination of the results obtained by these algorithms is then forwarded to the Scene Description module to finalize the overall scene understanding process.

3.4. Scene Description

The Scene Description module has two main responsibilities: enhance the analysis with semantic information to finalize the scene understanding process; provide means to describe the entire scene in a well-structured way. The internal structure of the module is divided into two levels. One for finalizing the scene interpretation and another to prepare the obtained data for distribution. To achieve these responsibilities, different methodologies can be considered when

we look at the specific requirements and constraints of a given application. For instance, relational databases could be used to store and manage scene data, which may be useful for applications that only require efficient retrieval and querying of data. Semantic networks [59], that represent objects or concepts as nodes, and semantic relationships between as edges are also useful as they can capture complex semantic relationships and dependencies within a scene. The ORA-SS (Object-Relationship-Attribute Data Model for Semi-structured Data) data model [60], which focuses on representing objects, their attributes, and the relationships between them in a more tabular or relational format, can also be applicable for scenarios where the scene structure is less hierarchical. However, these methodologies have limitations if we consider arbitrary visual scenes and their inherent hierarchical structure that combine both visual and semantic data can be of use for other applications. Taking these limitations into consideration and the fact that we have a data orchestrator that provides a global ontology/knowledge base, we argue that incorporating scene graphs could be advantageous. The choice of using scene graphs comes from the fact that they are an hierarchical structure that captures the objects in the scene as nodes and their relationships as edges. Each node represents an object, and edges denote relationships between objects. Another important aspect is that scene graphs provide a compact and expressive representation that allows for rich scene understanding and reasoning.

The module starts by extracting object-level and contextual information from the output of the Scene Analysis module, which includes detected objects, their attributes (such as colour, size, and shape), their spatial relationships, their interconnections within the scene, and then formulates a scene graph with these components. To enhance the accuracy and completeness of the scene description, the module leverages the structured knowledge within the ontology/knowledge base to correct potential errors or inconsistencies that may arise from the object detection or relationship detection processes by using a semantic reasoner to infer logical consequences from the data. It is important to highlight that there exist automatic scene graphs generation methods, which target both the scene analysis and description at the same time. Our idea is to incorporate these algorithms in the system, as well as enhance the description with other data extracted from the scene analysis. Works such as [61, 62, 63], are good candidates for this approach, as they leverage spatial and temporal information to better formulate and generate accurate scene graphs. Another good example is presented in [58], where ontologies are used to refine the resulting scene graphs to better target specific application scenarios.

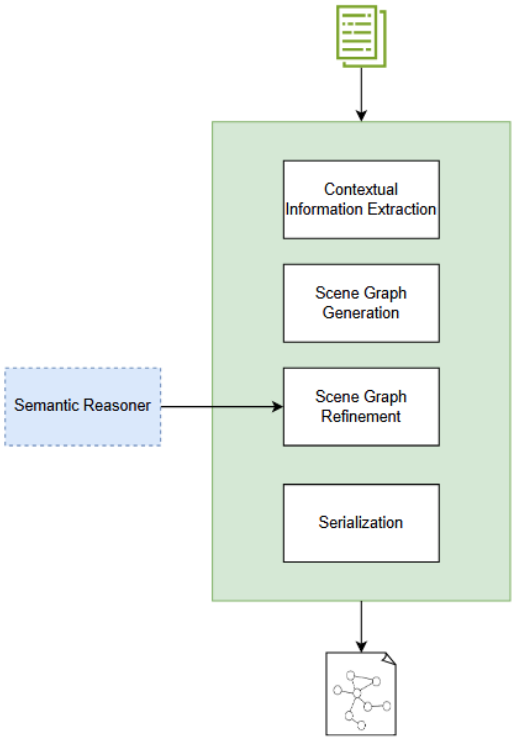

Figure 5: High level view of the scene description module. It receives the processed data from the scene analysis and extracts semantic information from it to then produce a scene graph. The output of the module is a serialized and refined graph.

After the generation of the scene graph and its refinement, the module prepares a data structure to output the graph in a readable and editable way. This involves encoding the generated scene graph and associated visual and semantic scene information into a suitable format that can be edited and interchanged, thus ensuring that the information obtained from the analysis of the scene can be stored and transmitted. A depiction of this module can be seen in Figure. 5. To create this textual representation, the module may utilize various data formats such as JSON, XML or other formats that are tailored to the specific needs of the system. For instance, it may also support a binary format that allows further information compression and faster read speeds. It is important that the chosen format ensures that the data is stored efficiently, and that the scene information extracted and processed is preserved both in integrity and completeness, so that this information can be further used without the need to analyse the scene once again.

Overall, this module has a pivotal role in this architecture, as it is crucial for providing a structured and semantically rich output of the complete analysis of a visual scene. The representation provided by the module also serves as a fundamental basis for further processing, visualization and storage or exchange of scene information. For instance, users can manually change the description to modify the underlying scene and therefore enable the synthesis of different scenes without changing anything in the processing pipeline. This capacity also enables the support of a wide range of application and tasks, by leveraging the rich information contained in the representation.

3.5. Scene Synthesis

The Scene Synthesis module is the final component of our architecture, and it takes as input the structured description of the scene provided by the previous module. It leverages this description

in order to generate a visually and semantically coherent 3D representation of the described scene. To achieve this, the module should internally combine 3D modelling techniques, rendering algorithms, and synthesis methods to generate a realistic and comprehensive visual output.

To generate the virtual scene, the module follows a series of steps, that depend on both the scene description and the user input that specifies the desired output. In a first phase, it starts by instantiating 3D models of the objects and their respective attributes within the scene. To do so, it can either utilize 3D modelling techniques such as geometric modelling or voxel-based representations or even a template-based approach where a 3D basic 3D model of each component is fetched from a database of previously generated models. Then, these models are positioned and arranged on the virtual scene according to the spatial information that is presented on the description. Once the positioning and instantiation of the models is made, the module begins the rendering process, where the lighting, shading, and texture mapping is made to improve the quality of the virtual environment. Naturally, this process is also very dependent of the user input that is given, as different techniques can be applied depending on the desired outcome. For instance, ray tracing, simple rasterization, global illumination simulation, and many others are techniques that can be employed in this process in order to provide the desired realism. This shows that the architecture not only supports different types of components for the scene analysis but also for the scene synthesis. Thus, enabling different implementations depending on the application scenario. To further improve realism techniques such as generative adversarial networks (GANs) or Neural Radience Fields (NeRF) could also be applied.

The final output of the Scene Synthesis module is expected to be a visually appealing and coherent 3D representation of the analysed scene, that ensures that the semantic information presented is the same as the one obtained from the scene analysis. This virtual representation should closely resemble the real-world environment described in the scene representation and follow the user instructions that define what type of output is to be created. This synthesized also has an important feature, where the representation can be rendered from different viewpoints and can have varying levels of detail, enabling visualization, virtual reality experiences, or integration into other applications. Figure 6 depicts a high-level view of the scene synthesis module.

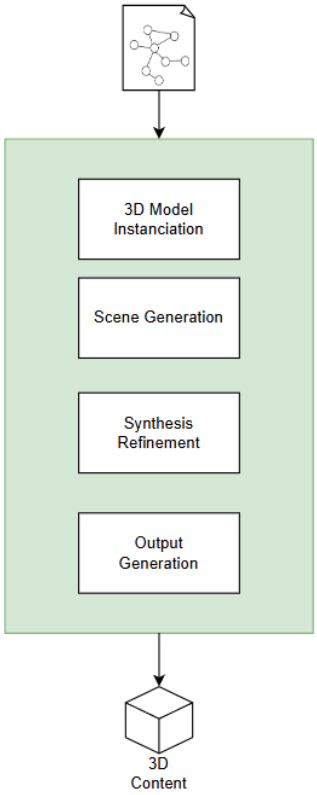

Figure 6: High level view of the scene synthesis module. It receives a serialized and refined graph and proceeds to translate the information present on the scene graph to 3D content that is then exported to a desired format.

The type and format of the output will also depend on specified user/application requirements and can be of the following forms: rendered images, rendered videos, 3D models, USD, or other application dependent data formats. It could also export 3D models of the synthesized scene in various formats, such as FBX (Filmbox), OBJ (Wavefront Object), or COLLADA (COLLAborative Design Activity), that may include the geometry, textures, materials, and animations of the scene, allowing for further manipulation or integration with other software. Similarly, the usage of USD could also be enabled as it is an open and scalable interchange format for 3D scenes that enables efficient storage, sharing, and collaboration across different software tools and platforms. Finally, this architecture could also enable the integration of other different data formats that could be specifically tailored to other applications for a seamless integration, or even allow the generation of mixed reality scenes, where virtual components are included into a real scene.

4. POSSIBLE USE CASES

As we have seen above, our proposal puts forward the building blocks necessary to create a system or framework that enables visual scene understanding description and possible posterior 3D scene representation. As the workflow transitions from scene analysis to representation and then to synthesis, it can be applied in multiple areas that are related to each of these steps. In this section we will present some potential applications and use cases for a system that implements our architecture, discussing what can be achieved.

As the last step of our proposal is the synthesis of virtual scenes, it is natural that one of the most direct use cases is related to content creation. By taking advantage of the possibility to pick a base scene and changed its description according to a user's desires, it is possible to synthesize 3D content for various purposes, such as movies, advertisements, or virtual worlds. Enabling content creators to generate realistic objects and scenes and without relying solely on physical setups or real-world recordings and manual labour creating the scenes from scratch. Furthermore, by having the possibility of controlling the synthesis process, it is possible to create different types of visualization of the same input data by enabling more or less detail or even providing different points of view. Similarly, it sees application in the creation and augmentation of datasets either of purely virtual spaces and actors, or mixed content, with virtual avatars in real scenes.

Another relevant area that sees usability is virtual and augmented reality, where users could recreate physical scenes and turn them into immersive and interactable virtual worlds, that could be easily changed by editing their associated descriptions. Additionally, by porting these descriptions into an augmented reality device, we could create virtual overlays with information of objects onto the real world, thus providing a seamless and immersing AR experience. In a similar fashion, it can also be applied for gamification or serious games, where specific situations can be recreated and used to provide valuable input for patient treatment and rehabilitation.

Another important area that sees advantages in using our proposal is in surveillance and overall security environments, where a complete visual scene understanding system for synthesis can be a great tool to provide multiple types of information. For instance, it could: detect anomalies by analysing the detections of objects, people and their interactions; provide a synthesized representation of the space, so that events can be analysed without infringing personal data laws, as only virtual representation of arbitrary human models is stored and provided; help better define the position of camera solutions to ensure better surveillance; provide a powerful tool to store and visualize past recordings without the need to store the original video.

By employing this architecture, we gain the ability to analyse, describe, and synthesize visual scenes with a high level of accuracy and realism that is derived from the applied algorithms. This enables us to understand the content of images and videos and represent scenes in a structured manner, and also providing flexible output options, allowing users to select the most suitable format for their needs. This way, it sees applicability in a wide range of applications encompassing multiple scenarios.

5. PROOF OF CONCEPT

In this section, we present a practical application of the proposed architecture for visual scene understanding and 3D synthesis, applied to a real-world scenario. In this implementation we target the specific problem of generation of hybrid data, that integrates real live scenes with virtual avatars performing actions based on real actors. To do so we detect and swap existing humans in videos with 3D avatars performing the same activities, ensuring the anonymity of the original actors, which is an important aspect when data protection specifications are highly restrictive. This system integrates our proposed architecture and makes use of state-of-the-art technologies. In this implementation we show how each component of the system can be plugged, highlighting the seamless integration of the scene analysis, description, and synthesis modules. As this proof of concept intends to show the possibilities of the proposed architecture, we do not elaborate a complex ontology or define multiple rules. Rather, we focus on detecting, tracking and translating humans from videos to a virtual representation that is then synthesized for an output video, while also allowing manual change of the description in order to manipulate the outcome of the synthesis without interfering with the input data.

In the scene analysis module we detect, segment and track humans in the input videos. This is achieved by relying on the state-of-the-art YOLO-v8, presented by Ultralytics [64]. This way, we identify bounding boxes, segmentation masks and obtain IDs for each human in the scene. This information is then provided to the VIBE [65] pose estimation algorithm, so that the pose of each human in the scene is detected and mapped to SMPL models, which are versatile and realistic representation of human body shape and pose [66]. For the scene description, we formulate a simple scene graph that contains the corresponding ID of the associated person, as well as pose and SMPL data associated with them, for each frame. In this implementation we also store separately the corresponding segmentation masks that can be used to aid the inpainting process required to remove the original detections from the images. This information is then serialized to a JSON file, as we want to illustrate the ability to store and distribute information extracted from the analysis of the scenes. This also shows the ability of the system to divide the processing pipeline, where Scene Analysis and Description can first be performed, and the Scene Synthesis module can later use the generated description to generate the synthesis. For the rules established in this implementation, we simply select an inpainting process if the user wishes to swap the humans in the input video with the 3D avatars. Otherwise, the avatars are overlapped in the resulting video. For the inpainting we used the E^2FGVI algorithm [67] as it shows impressive results. To illustrate the results obtained by our proof of concept with and without inpainting, we selected videos from the HMDB51 dataset [68].

In Figure 7 we show a simple example of the possible outcomes that can be obtained by our implementation. We can place avatars over the original video or remove the original actor and place an avatar on its place.

Figure 7: Example of a video of a woman swinging a golf club. In the center we have an avatar overlaping the original actor and in the right we completely remove the actor.

In the next example, depicted in Figure 8, we show that for an arbitrary frame we can obtain a corresponding scene graph that represents the information in it. Furthermore, by manipulating the graph and introducing a rotation attribute, we can tell the Scene Synthesis module to apply a rotation on the generation of the avatar. In the graph, we use the special +has relationship to include the data extracted from the analysis.

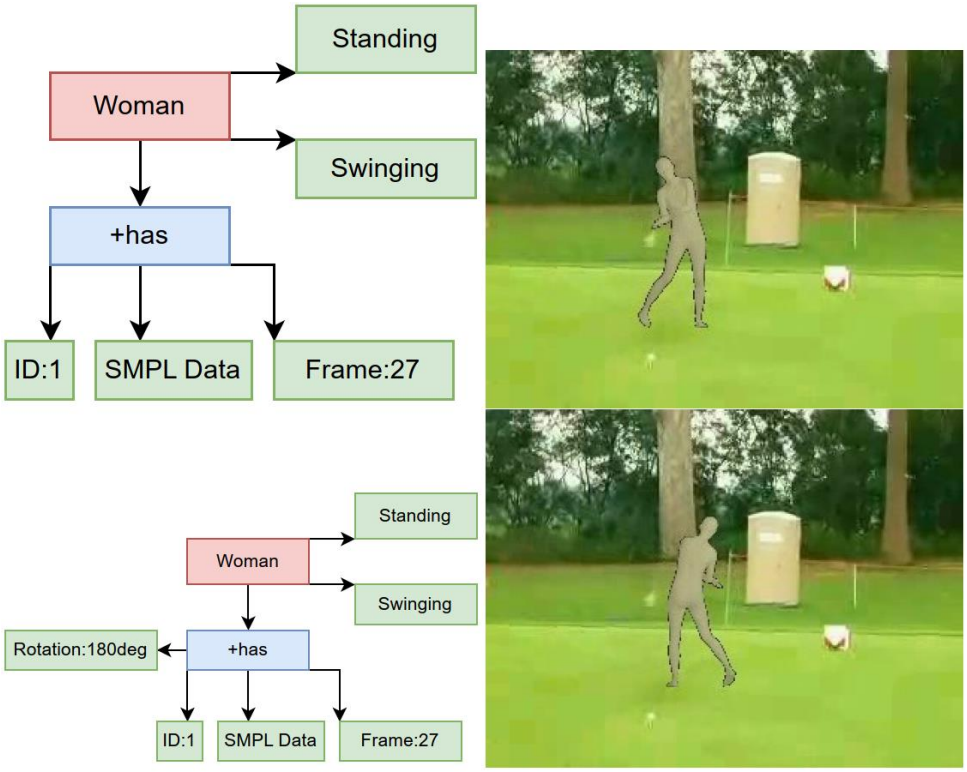

Figure 8: In the top images we have an avatar swinging a golf club, with the corresponding scene graph associated. On the bottom we see the same synthesis but with a modification in the scene graph, resulting in a rotation of the avatar on the synthesis process. We use red to denote entities, green for attributes and blue for relationships

We are also able to generate hybrid content that is semantically different from the original video. For instance, in Figure 9 we show that we can transform a video of a person running towards a pier in a video where the avatar is actually running from the pier. We are able to do this by introducing a rotation and specifying to the Scene Synthesis module to produce the results backwards.

Figure 9: Example of a video of a woman running towards a pier on the top three images. In the bottom we have an avatar rotated 180 degrees and running from the pier, as we have generated the synthesis backwards.

6. CONCLUSION AND FUTURE WORK

In this work we explore the vast field that is visual scene understanding and introduce a different paradigm that is visual scene understanding for 3D synthesis, where visual and semantic data extracted from real visual scenes can be leveraged to provide concise descriptions of their underlying scenes and used to recreate 3D virtual environments. We further explore this idea and present a comprehensive architecture for the task of visual scene understanding for 3D synthesis. We address the complexities of analysing, describing, and recreating scenes in 3D through the proposal of a modular architecture that allows a seamless integration of scene analysis, description, and synthesis.

For each module of our proposal, we explore existing technologies and algorithms that can be employed and show how each module can interact with the other to ensure that the information is always well understood inside all of the modules. We also highlight the benefits of this approach and explore real-world applications where our proposal can have impact. Additionally, we provide a proof-of-concept example where the architecture's ability to detect and track humans, estimate and describe their poses, and generate 3D avatars is demonstrated. We combine state-of-the-art computer vision techniques and algorithms to show that it is possible to use our proposal to generate and manipulate visual content with 3D virtual humans performing actions. Compared to traditional methodologies, our proposal enables a unique architecture that can be leveraged for multiple workflows that require the processed data obtained from images or videos. Thus, also enabling a faster development process for new applications.

While our work provides a robust foundation for the development of a system or framework, it leaves multiple opportunities for future research and development of associated topics, namely: research on fine-grained scene description supported by the scene graphs that allow precise details about object attributes, environment, scene illumination or group behaviours, which can lead to even more precise descriptions of the scene and ultimately a better and more realistic scene synthesis; Dynamic scene synthesis, where the generated content can incorporate lighting changes; real-time processing for integration in applications such as live broadcasts or mixed reality scenarios. Naturally, other challenges and research opportunities could also be

encountered when applying our proposal into more specific application scenarios, ultimately showing that this is a new and challenging task that can see a broad applicability.

ACKNOWLEDGEMENTS

The authors would like to thank everyone, just everyone! The work was funded by the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101094831 (Project Converge – Telecommunications and Computer Vision Convergence Tools for Research Infrastructures). Américo was funded by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within the PhD grant SFRH/BD/146400/2019.

REFERENCES

[1] X. Cui, D. Khan, Z. He, Z. Cheng, Fusing surveillance videos and three dimensional scene: A mixed reality system, Computer Animation and Virtual Worlds 34 (1) (2023) e2129.

[2] A. Pereira, P. Carvalho, N. Pereira, P. Viana, L. Côrte-Real, From a visual scene to a virtual representation: A cross-domain review, IEEE Access 11 (2023) 57916–57933. doi:10.1109/ACCESS.2023.3283495.

[3] A. Brock, S. De, S. L. Smith, K. Simonyan, High-performance large-scale image recognition without normalization, in: International Conference on Machine Learning, PMLR, 2021, pp. 1059–1071.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

[5] X. Ding, C. Xia, X. Zhang, X. Chu, J. Han, G. Ding, Repmlp: Reparameterizing convolutions into fully-connected layers for image recognition, arXiv preprint arXiv:2105.01883 (2021).

[6] H. Wang, X. Jiang, H. Ren, Y. Hu, S. Bai, Swiftnet: Real-time video object segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1296–1305.

[7] L. Hu, P. Zhang, B. Zhang, P. Pan, Y. Xu, R. Jin, Learning position and target consistency for memory-based video object segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4144–4154.

[8] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, R. Yang, Salient object detection in the deep learning era: An in-depth survey, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).

[9] L. Qin, Y. Shi, Y. He, J. Zhang, X. Zhang, Y. Li, T. Deng, H. Yan, Id-yolo: Real-time salient object detection based on the driver's fixation region, IEEE Transactions on Intelligent Transportation Systems (2022).

[10] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv preprint arXiv:1804.02767 (2018).

[11] P. Hurtik, V. Molek, J. Hula, M. Vajgl, P. Vlasanek, T. Nejezchleba, Poly-yolo: Higher speed, more precise detection and instance segmentation for yolov3, Neural Computing and Applications (2022) 1–16.

[12] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, L. Fei-Fei, Image retrieval using scene graphs, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3668–3678.

[13] L. Gao, B. Wang, W. Wang, Image captioning with scene-graph based semantic concepts, in: Proceedings of the 2018 10th International Conference on Machine Learning and Computing, 2018, pp. 225–229.

[14] X. Yang, K. Tang, H. Zhang, J. Cai, Auto-encoding scene graphs for image captioning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 10,685–10,694.

[15] D.-J. Kim, J. Choi, T.-H. Oh, I. S. Kweon, Dense relational captioning: Triple-stream networks for relationship-based captioning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6271–6280.

[16] R. Koner, H. Li, M. Hildebrandt, D. Das, V. Tresp, S. G¨unnemann, Graphhopper: Multi-hop scene graph reasoning for visual question answering, in: International Semantic Web Conference, Springer, Cham, 2021, pp. 111–127.

[17] S. V. Nuthalapati, R. Chandradevan, E. Giunchiglia, B. Li, M. Kayser, T. Lukasiewicz, C. Yang, Lightweight visual question answering using scene graphs, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 3353–3357.

[18] A. Cherian, C. Hori, T. K. Marks, J. Le Roux, (2.5+ 1) d spatiotemporal scene graphs for video question answering, arXiv preprint arXiv:2202.09277 (2022).

[19] M.-D. Nguyen, B. T. Nguyen, C. Gurrin, A deep local and global scenegraph matching for image-text retrieval, arXiv preprint arXiv:2106.02400 (2021).

[20] B. Schroeder, S. Tripathi, Structured query-based image retrieval using scene graphs, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 178–179.

[21] S. Mody, J. Thakkar, Analysis of image generation using scene graphs, in: 5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD), 2022, pp. 296–297.

[22] Y. Xue, Y.-C. Guo, H. Zhang, T. Xu, S.-H. Zhang, X. Huang, Deep image synthesis from intuitive user input: A review and perspectives, Computational Visual Media 8 (1) (2022) 3–31.

[23] Y.-H. H. Tsai, S. Divvala, L.-P. Morency, R. Salakhutdinov, A. Farhadi, Video relationship reasoning using gated spatio-temporal energy graph, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 10,424–10,433.

[24] A. V. Malawade, S.-Y. Yu, B. Hsu, D. Muthirayan, P. P. Khargonekar, M. A. Al Faruque, Spatio-temporal scene-graph embedding for autonomous vehicle collision prediction, IEEE Internet of Things Journal (2022).

[25] J. Johnson, A. Gupta, L. Fei-Fei, Image generation from scene graphs, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1219–1228.

[26] M. Bilal, M. Maqsood, S. Yasmin, N. U. Hasan, S. Rho, A transfer learning-based efficient spatiotemporal human action recognition framework for long and overlapping action classes, The Journal of Supercomputing 78 (2) (2022) 2873–2908.

[27] R. D. Brehar, M. P. Muresan, T. Marit¸a, C.-C. Vancea, M. Negru, S. Nedevschi, Pedestrian street-cross action recognition in monocular far infrared sequences, IEEE Access 9 (2021) 74302–74324.

[28] S. Patil, K. S. Prabhushetty, A survey on human action recognition and detection techniques, in: ICT Analysis and Applications, Springer, Singapore, 2022, pp. 157–165.

[29] L. Song, G. Yu, J. Yuan, Z. Liu, Human pose estimation and its application to action recognition: A survey, Journal of Visual Communication and Image Representation 76 (2021) 103055.

[30] B. Ren, M. Liu, R. Ding, H. Liu, A survey on 3d skeleton-based action recognition using learning method, arXiv preprint arXiv:2002.05907 (2020).

[31] W. Zhou, S. Dong, J. Lei, L. Yu, Mtanet: Multitask-aware network with hierarchical multimodal fusion for rgb-t urban scene understanding, IEEE Transactions on Intelligent Vehicles 8 (1) (2023) 48–58. doi:10.1109/TIV.2022.3164899.

[32] A. Hu, F. Cotter, N. Mohan, C. Gurau, A. Kendall, Probabilistic future prediction for video scene understanding, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16, Springer, 2020, pp. 767–785.

[33] W. Hu, H. Zhao, L. Jiang, J. Jia, T.-T. Wong, Bidirectional projection network for cross dimension scene understanding, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14373–14382.

[34] Y. Cai, X. Chen, C. Zhang, K.-Y. Lin, X. Wang, H. Li, Semantic scene completion via integrating instances and scene in-the-loop, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 324–333.

[35] A.-Q. Cao, R. de Charette, Monoscene: Monocular 3d semantic scene completion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 3991–4001.

[36] Y. Nie, J. Hou, X. Han, M. Niessner, Rfd-net: Point scene understanding by semantic instance reconstruction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 4608–4618.

[37] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, C. Stachniss, Towards 3d lidar-based semantic scene understanding of 3d point cloud sequences: The semantickitti dataset, The International Journal of Robotics Research 40 (8-9) (2021) 959–967.
[38] K. Wang, G. Zhang, J. Yang, 3d human pose and shape estimation with dense correspondence from a single depth image, The Visual Computer (2022) 1–13.
[39] T. Xu, Y. Fujita, E. Matsumoto, Surface-aligned neural radiance fields for controllable 3d human synthesis, arXiv preprint arXiv:2201.01683 (2022).
[40] R. Vidaurre, I. Santesteban, E. Garces, D. Casas, Fully convolutional graph neural networks for parametric virtual try-on, Computer Graphics Forum 39 (8) (2020) 145–156.
[41] H. Yu, C. Cheang, Y. Fu, X. Xue, Multi-view shape generation for 3d human-like body, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) (2022).
[42] U. Technologies, Unity3d, https://unity.com, accessed on 27 September 2023 (2023).
[43] I. Epic Games, Unreal engine 4, https://www.unrealengine.com, accessed on 27 September 2023 (2023).
[44] B. Foundation, Blender, https://www.blender.org, accessed on 27 September 2023 (2023).
[45] A. Baillet, E. Murphy, O. Dunn, M. Gao, Forging a new animation pipeline with usd, in: ACM SIGGRAPH 2018 Talks, 2018, pp. 1–2.
[46] A. Ferreira, J. Li, K. L. Pomykala, J. Kleesiek, V. Alves, J. Egger, Ganbased generation of realistic 3d data: A systematic review and taxonomy, arXiv preprint arXiv:2207.01390 (2022).
[47] K. Fu, J. Peng, Q. He, H. Zhang, Single image 3d object reconstruction based on deep learning: A review, Multimedia Tools and Applications 80 (2021) 463–498.
[48] Z. Shi, S. Peng, Y. Xu, Y. Liao, Y. Shen, Deep generative models on 3d representations: A survey, arXiv preprint arXiv:2210.15663 (2022).
[49] K. Gao, Y. Gao, H. He, D. Lu, L. Xu, J. Li, Nerf: Neural radiance field in 3d vision, a comprehensive review, arXiv preprint arXiv:2210.00379 (2022).
[50] A. Rabby, C. Zhang, Beyondpixels: A comprehensive review of the evolution of neural radiance fields, arXiv preprint arXiv:2306.03000 (2023).
[51] B. McBride, The resource description framework (rdf) and its vocabulary description language rdfs, in: Handbook on ontologies, Springer, 2004, pp. 51–65.
[52] G. Antoniou, F. v. Harmelen, Web ontology language: Owl, Handbook on ontologies (2009) 91–110.
[53] M. Sporny, D. Longley, G. Kellogg, M. Lanthaler, N. Lindstr¨om, Json-ld 1.1, W3C Recommendation, Jul (2020).
[54] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.
[55] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, A. Veit, et al., Openimages: A public dataset for large-scale multi-label and multi-class image classification, Dataset available from https://github. com/openimages 2 (3) (2017) 18.
[56] J. I. Olszewska, T. L. McCluskey, Ontology-coupled active contours for dynamic video scene understanding, in: 2011 15th IEEE International Conference on Intelligent Engineering Systems, IEEE, 2011, pp. 369–374.
[57] F. K. Kenfack, F. A. Siddiky, F. Balint-Benczedi, M. Beetz, Robotvqa—a scene-graph-and deep-learning-based visual question answering system for robot manipulation, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 9667–9674.
[58] F. Amodeo, F. Caballero, N. Díaz-Rodríguez, L. Merino, Og-sgg: ontology-guided scene graph generation—a case study in transfer learning for telepresence robotics, IEEE Access 10 (2022) 132564–132583.
[59] J. F. Sowa, et al., Semantic networks, Encyclopedia of artificial intelligence 2 (1992) 1493–1511.
[60] G. Dobbie, T. W. Ling, Object relationship attribute data model for semi-structured data, in: Encyclopedia of Database Systems, Springer US, 2009, pp. 1940–1941.
[61] Y. Cong, W. Liao, H. Ackermann, B. Rosenhahn, M. Y. Yang, Spatial-temporal transformer for dynamic scene graph generation, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 16372–16382.
[62] L. Xu, H. Qu, J. Kuen, J. Gu, J. Liu, Meta spatio-temporal debiasing for video scene graph generation, in: European Conference on Computer Vision, Springer, 2022, pp. 374–390.

[63] Y. Li, X. Yang, C. Xu, Dynamic scene graph generation via anticipatory pre-training, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 13874–13883.
[64] G. Jocher, A. Chaurasia, J. Qiu, Yolo by ultralytics, gitHub repository (January 2023). https://github.com/ultralytics/ultralytics
[65] M. Kocabas, N. Athanasiou, M. J. Black, Vibe: Video inference for human body pose and shape estimation, in: European Conference on Computer Vision (ECCV), 2020.
[66] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, M. J. Black, Smpl: A skinned multi-person linear model, in: ACM Transactions on Graphics (TOG), Vol. 34, ACM, 2015, pp. 248:1–248:16.
[67] Z. Li, C.-Z. Lu, J. Qin, C.-L. Guo, M.-M. Cheng, Towards an end-to-end framework for flow-guided video inpainting, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022.
[68] H. Kuehne, H. Jhuang, R. Stiefelhagen, T. Serre, Hmdb: A large video database for human motion recognition, in: 2011 International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 2556–2563.

AUTHORS

Américo Pereira is a researcher at INESC TEC since 2014 and Ph.D. student in electrical and computer engineering at the University of Porto since 2015. He holds a M.Sc. degree in computer science from the same university (2011, 2013). His work focuses on computer vision, image/video processing, and machine learning.

Pedro Carvalho is a senior researcher at INESC TEC since 2001, with a Ph.D. in electrical and computer engineering (2012) and M.Sc. in network and communication services (2004) from the University of Porto. He is also an adjunct professor at the Polytechnic Institute of Porto since 2014. His research interests include image/video processing and computer vision.

Luís Côrte-Real is an associate professor at the University of Porto, with a Ph.D. in electrical engineering (1994) and an M.Sc. in electrical and computer engineering (1986). He is also a Researcher at INESC TEC since 1985, focusing on image/video processing and coding.