Multimedia Tools and Applications
https://doi.org/10.1007/s11042-023-16169-0

View recommendation for multi-camera demonstration-based training

Saugata Biswas[1] · **Ernst Kruijff**[1] · **Eduardo Veas**[2]

Received: 13 March 2023 / Revised: 14 June 2023 / Accepted: 1 July 2023

Abstract
While humans can effortlessly pick a view from multiple streams, automatically choosing the best view is a challenge. Choosing the best view from multi-camera streams poses a problem regarding which objective metrics should be considered. Existing works on view selection lack consensus about which metrics should be considered to select the best view. The literature on view selection describes diverse possible metrics. And strategies such as information-theoretic, instructional design, or aesthetics-motivated fail to incorporate all approaches. In this work, we postulate a strategy incorporating information-theoretic and instructional design-based objective metrics to select the best view from a set of views. Traditionally, information-theoretic measures have been used to find the goodness of a view, such as in 3D rendering. We adapted a similar measure known as the viewpoint entropy for real-world 2D images. Additionally, we incorporated similarity penalization to get a more accurate measure of the entropy of a view, which is one of the metrics for the best view selection. Since the choice of the best view is domain-dependent, we chose demonstration-based training scenarios as our use case. The limitation of our chosen scenarios is that they do not include collaborative training and solely feature a single trainer. To incorporate instructional design considerations, we included the trainer's body pose, face, face when instructing, and hands visibility as metrics. To incorporate domain knowledge we included predetermined regions' visibility as another metric. All of those metrics are taken into account to produce a parameterized view recommendation approach for demonstration-based training. An online study using recorded multi-camera video streams from a simulation environment was used to validate those metrics. Furthermore, the responses from the online study were used to optimize the view recommendation performance with a normalized discounted cumulative gain (NDCG) value of 0.912, which shows good performance with respect to matching user choices.

Keywords Multi-camera · Camera view analysis · View selection · Camera selection · Demonstration-based training · Instruction design · Entropy · Recommender systems

✉ Saugata Biswas
saugata.biswas@h-brs.de

Extended author information available on the last page of the article

Published online: 03 August 2023

1 Introduction

Learning is an important aspect of everyday life for acquiring new skills, for example, through some form of instruction or teaching. A typical example of teaching is demonstration-based training (DBT), where students learn through observation. DBT scenarios are usually face-to-face meetings, for example in a classroom setting. However, other types of DBT have become popular, from interactive manuals to video recordings. For example, millions of instructional videos can be found online on platforms such as YouTube. Ideally, when co-located, trainees can communicate with and observe a trainer in a face-to-face manner, supporting direct communication. Flexibility to change viewpoints in a such face-to-face situation by simply moving around physically – a typical action in real-world classrooms – is another advantage. This allows for resolving visual ambiguities and invariances [99, 107]. Yet, especially instruction videos do not support this flexibility. Viewpoints are typically generated by a single camera, which is usually fixed or worn on the hand or head. Consequently, arbitrary viewpoints are not supported. This may cause typical issues such as occlusions (e.g., the instructor occluding the actual object of interest) or a lack of close-up of certain actions. To solve this, a possible solution is the deployment of multi-camera systems. Such systems integrate strategically placed static cameras that can be complemented by video streams from dynamic viewpoints, e.g., a camera connected to the head. Using a front-end such as a web interface, users can potentially select the best view from all offered views to view the instruction in an optimal manner. However, especially when the number of cameras increases, this likely is not ideal: trainees would need to continuously monitor multiple camera feeds quite similar to security monitoring systems [89] while paying attention to the instructions. This may be counterproductive for the learning process as attention (cognitive) resources are limited [67]. Hence, presenting users with the best view or views among a set of views could be advantageous to reduce issues associated with attention and limited cognitive resources. Unfortunately, current literature is not conclusive as to what parameters need to be regarded to select an optimal view from a set of views during a training session. Though literature exists in the domain of cinematography, observational learning, and illustrative rendering [37, 54], there is no universally accepted definition of what constitutes the "best view". An overview of these sources can be found in the related work (Section 2) - we derive requirements or guidelines for our system where possible. In related work, views are often judged by aesthetics or they are ranked depending on viewers' choices in a specific scenario, rather than a set of objective metrics, which is not relevant to DBT scenarios.

Therefore, it is essential to identify these objective metrics. The best view could also be dependent on modeling or prior knowledge of the information content distribution within the DBT scene. If the scenario changes significantly, the model would need to be updated. Moreover, it might not always be possible to reliably model the scene and underlying semantics. Hence, a parameterized approach for view selection is desired. However, prior information about the scene should not be discarded completely as it could be useful for view selection. Prior knowledge should be incorporated in the best view selection process.

Best views are intended to be shown as a larger image in a web front-end while showing the rest of the streams as thumbnails. The focus of this paper is on the underlying system, our recommender system. The front end itself is outside our scope. The recommender system is driven by image analysis to select two feature views: an action view that is optimized to view the instructor and operation space and a communication view that shows the face of the instructor to support direct communication. The action and communication views can be incorporated into a single view that best represents the scene by optimizing all underlying

factors. For a group of objects of interest in a scene, the best view can be defined as the one that provides maximum visibility of the objects, contains most of the information about the objects, and shows the objects as magnified as possible. For a DBT scenario, the best view should also show the actions (including gestures) performed by the trainer in the most informative way, defined by a set of metrics predominantly related to the trainer's actions. Namely, in our system, the best views are selected by performing different types of image analysis related primarily to the trainer's activities. The system detects motion, the visibility of body parts, and regions of interest. We do so by calculating the entropy of an image – being an entity to quantify average information content in a view – by feeding in the metrics of the image analysis. The Entropy of an image expresses the average information or uncertainty of its building blocks or the pixels of the image, evaluating the informativeness of the view and providing the system a measure as to which view to select as the best view. Instead of solely relying on prior knowledge about the scene, we use a view ranking algorithm that considers current information content in the scene. However, not all prior knowledge regarding the training scenario should be discarded. It can be defined as a region of interest, a concept we also use in our system.

In this paper, we describe the technical system and motivation for the different analyses that form the basis for the entropy calculation. We discuss what potentially makes up the best view, and how different analyses can contribute to its calculation. To address if the best view recommendation matches what users would select, we performed an online user study, where 43 persons manually tagged the best view from a set of 8 views on a scene and rated all views individually. We developed a view recommender (explained in Section 5) for the multi-camera system that automatically selects and presents the best view. We designed the view recommender to be parameterized, allowing for easy adaptation to varying conditions in DBT. We compared the ranked lists from the user study with the views selected by our recommender system, and closely analyzed the role of different types of analysis through user feedback. The outcomes revealed that our first system iteration matched the view selection by users well in most cases: a normalized discounted cumulative gain (NDCG) score of 0.906 was achieved. View selection matched the user's topmost selection 42.4% of the times it was used. By adjusting the weights based on user feedback, we ran a further iteration, improving the NDCG score to 0.912. To our recollection, this is the first automated view recommendation system for a multi-camera system used in DBT that deploys carefully selected image analysis.

2 Related work

Development of our view recommender system draws from prior work, which we outline below.

2.1 Instruction and learning systems

An instructional video is a form of multimedia where targeted instructional material is delivered using dynamic graphics, i.e., videos and words in the form of speech and background sound [12, 33]. In DBT, the viewers learn by observing a trainer during task performance. This type of learning, i.e., learning by visualization of the demonstration being enacted, is called observational learning [10, 11, 97]. In the case of imitation learning, viewers watch the relative motion of the demonstrator's limbs [45, 47]. Thereby, sensorimotor observational learning can lead to learning in complex tasks [17, 40, 124]. An instructional video can facili-

tate observational learning [14, 116]. Although the use of instructional videos for educational purposes is widely used [25, 101, 110], studies generally do not focus on DBT scenarios. Instructional design theories, describing how to design instructional videos [73, 84, 85, 96] have been reported on, yet do not necessarily give any guidelines for camera view or angle setting. Although some approaches mention mixing first-person and third-person camera views [33, 74], these studies do not offer any heuristics for camera positioning. Instructional video design guidelines offer some insight into how to present the instructor [73]. Some research has focused on if the instructor should be shown, what part of the instructor should be shown, and how it affects different aspects of learning [43]. The presence of the instructor can influence students' engagement and learning while seeing the instructor talking engages viewers more than slides [69]. It has been shown that the instructor's face visibility allocates a substantial amount of students' attention [72, 119, 120]. However, there are mixed opinions if it does [69] or does not [33] aid learning. A static instructor face can be detrimental to the learning experience, whereas a gesturing instructor pose helps to learn [73]. Also, the instructor's eye gaze has a stronger influence than body orientation on students' attention and learning [88]. Hence, it would be beneficial to consider the instructor's face visibility, especially when they are giving vocal instructions, and their body pose in the best view.

2.2 Multi-camera systems

Multi-camera systems have been used in surveillance, education, sports, and mobile systems [82]. Multi-camera systems can offer more than one view of a scene at a time. This advantage can potentially mitigate some issues with single view systems, e.g., poor camera placement or narrow field of view [19, 64]. To maximally cover the scene area cameras need to be placed optimally. Cameras can be placed with heuristic reasoning [68], maximum coverage [18], and by avoiding dynamic occlusion [90]. Some multi-camera-equipped lecturing systems have been developed [2, 5, 63, 77, 112]. These systems usually include lecture material like slides or boards. Multi-camera instructional videos can be more valuable and engaging compared to single camera-based instructions [117]. Furthermore, a study on learning medical hand procedures showed that multi-camera setups have maximized training effect [16]. Most systems are concentrated on the lecturer moving in the classroom. Although systems are capable of capturing the best region of interest depending on lecturer movement [5], audience activity, or focus of lecture materials, those are not built for detailed procedural knowledge transfer in a DBT scenario. For example, in [41] two pan-tilt-zoom cameras are placed in a classroom that offers views such as room overview, speaker view, audience overview, and questioner view. A screen view is also added by capturing the presentation screen. All of these views were considered as states in a finite state machine. View switching is done by considering predefined events (e.g., movement or speech detection) or timeouts. However, it only detects changes to switch states and does not consider information content in the view for differentiation. Our system is different from this system because we are using multi-view from multi-cameras and do not differentiate objective-wise between cameras. Some approaches try to capture the moving human from the best possible viewpoint by capturing limb movements. For example, drone cinematography for action scenes can be based on maximum variance (by eigenvalue decomposition) of body skeleton landmarks [133]. In due course, a multi-camera view can help with imitation learning [57]. Note we further reflect upon activity analysis in Section 3.1.

2.3 Recommender systems

Recommender systems are information filtering systems that predict the preference for an item of a user. Its main objective is to suggest relevant items to users. Recommender systems are widely used in e-commerce [4, 132], social media [131], entertainment [98], and other domains [113]. In e-commerce, it is used to recommend relevant products for a user's search, e.g., Amazon, eBay, etc. In social networks, it is used to suggest connections between users. Recommendation systems are used to generate playlists for video and music streaming services, e.g., Netflix, YouTube, Spotify, etc. Recommender systems are of mainly three types: content-based, collaborative filtering, and hybrid approaches [23, 60], which is a combination of the first two types. Content-based systems recommend the items based on a model created from past user-item interactions. Collaborative filtering approaches recommend items based on inter-users or inter-items rating profiles. By searching items or users in the neighborhood of the current user or item, they generate similar recommendations. In this work, we want to develop a recommender system for camera views in a multi-camera scenario. Although we call it a view recommender, it differs from traditional recommender systems because it recommends a single view from a limited set of views by rating them. We try to model the recommender system as a parameterized one that would potentially work for various scenarios by tuning the parameters (more of it is explained in Section 5).

3 Viewpoint analysis and selection methods

The camera view that offers the most informative and useful information about the scene can be labeled as the best view. Best view finding by viewpoint optimization has been explored in various domains, including cinematography and robotics. We reflect upon these domains to derive useful metrics for our best view selection methods.

3.1 Activity analysis

During DBT training, a trainer demonstrates a training procedure while instructing viewers how to perform the task at hand. These instructional DBT scenarios generally include different kinds of procedural tasks like assembly, operation, repair, or maintenance. [65, 130]. In an instructor-led instructional DBT scenario, usually, the trainer has all of the required equipment available. The trainer uses their hands to, for example, pick up or operate some equipment. These manipulations of objects are regarded as training actions. Sometimes these training actions involve partial or full body movement of the trainer [57, 124]. Tasks involving those actions can be categorized into three types: local, spatial, and body-coordinated [100]. Local tasks are single-handed steps performed by the trainer not changing their location in the scene. Spatial actions are done by moving from one place to another in the scene. Finally, body-oriented actions are performed by using their hands, body, and eye coordination in unison.

In DBT, actions are usually performed along with verbal instructions to explain actions or describe the strategy for the task. Sometimes this verbal form of communication is accompanied by non-verbal cues, for example, gestures. These include movement of the head, hand, or parts of the body [22]. Gestures also play an important role in collaborative training sessions [6, 30, 36, 106]. Among different types of gestures [75], pointing and representational gestures are important [36]. Pointing or deictic gestures involve the speaker pointing to an

object in the scene and verbally saying *this, that, here, there*, etc. On the other hand, representational or iconic gestures [123] include hand movements that show a perceptual relation with speech content. For example, a speaker might move their hand sideways to indicate a flat surface. Gestures - iconic, spatial, kinetic - play a key role in task-oriented dialogues [8, 81]. These gestures offer communication grounding [70], which helps to establish a common ground among the collaborators. Therefore, gestures realized by limb movements play an important role in communication during training. Moreover, the visibility of hands is more effective for learning manual manipulation tasks than just reading manuals [26, 91]. Hence, it is beneficial to consider the visibility of hands as an important metric for the best view.

There have been ample works for human activity analysis using computer vision techniques [3, 13, 86]. Although we can identify human activities from video, we did not want to focus on any specific activities. In this work, we relied upon only human pose estimation [21, 24, 78] so that we can separate the head, body, and limbs from the video stream.

3.2 Entropy

With regards to view analysis, information-theoretic approaches for the evaluation of the goodness of a viewpoint have been explored based on the notion of information measure or entropy [109]. Since entropy quantifies the average information content of a variable, it can be extended to evaluate information associated with a viewpoint. This concept is known as viewpoint entropy [37], which is also a key aspect of our approach. Viewpoint entropy has been measured by projecting the visible surface areas of an object on a tessellated sphere centered on the viewpoint. Viewpoint entropy gives a measure of how much of an object's surface polygons are seen from the associated viewpoint and has been used to select the best viewpoints, e.g., for image rendering [50, 126, 127, 129, 134]. The idea of viewpoint entropy was extended by adding chrominance, luminance, and weighted object priority to understand a scene [125]. A relative entropy or Kullback-Liebler (KL) distance-based viewpoint optimization [122] and integration with mesh saliency [29], silhouette and depth attributes [108], object uniqueness [35] were also introduced. Recent polygon-based view selection surveys include more of these model-based approaches [15, 49]. Although these methods yielded good results for 3D models, it was not shown how to extend these methods for comparison of real-world scenarios. Of further interest are information-theoretic measures that have also been used for comparison of images [61] and shot cut/fade detection in video sequences [20]. Although classical entropy has been used for viewpoint optimization [56, 111, 121], it suffers from a major weakness. The classical definition of entropy has one inherent weakness: it does not consider the spatial distribution of the inputs. Thereby, it cannot measure the true information content of an image. Some have tried to overcome this weakness [92, 105]. However, these methods are computationally expensive compared to the classical implementation of entropy measures. It should be mentioned that viewpoint entropy [37] considers pixels' spatial relationship to some extent because it groups neighboring pixels that belong to the same face on an object. Since models of objects are not always known, model-less approaches for viewpoint selection have been developed [39, 42, 62, 118] in the context of object recognition.

3.3 Views in virtual cinematography and robotics

Virtual cinematography focuses on aesthetics, cinematographic elements, emotion capture, and group interactions that capture for example dialogue and gaze for movies, animations, and

games. However, these approaches are not necessarily focused on humans interacting with objects. For these reasons, the virtual cinematography approaches are often not adequate for DBT. However, they could be useful for framing and visual element placement, by considering best production practices. In virtual cinematography, view selection or framing has been based on continuity of editing by considering stylistic rules [59], visibility of key subjects [76], user preferences [7], and topology awareness [1]. Inspired by filmmaking practices, constraint-based editing tools help editors, for example, to pick the best camera positions, framing, actor's size on screen, or desired visual features [31]. Example-driven approaches that can learn from sample movie clips and popular photos that reproduce the camera positioning for virtual cinematography have also been developed [9, 52, 102]. Instructional video design guidelines are mainly concentrated on instruction material presentation, showing a trainer in traditional classroom lectures or online courses. The guidelines do not yet consider how-to training tasks (e.g., assembly, repair, etc.) that show and explain the procedures step by step. Also, design guidelines do not clarify where the camera should be placed, which camera views should be selected, or how to switch between camera views – all of which are vital for how-to-do procedural DBT scenarios. In virtual cinematography, framing is based on aesthetics, emotion capture, and gaze estimation of the actors. Although close-up shots allow for better visibility, no guidelines are offered on how to manage the frame or view in DBT.

In **robotics**, viewpoint optimization has been used for object recognition, scene reconstruction, robot localization, visual inspection, and task planning. Earlier work posed viewpoint finding as an optimization problem for better viewing of some scene objects' geometrical features, which required prior knowledge (e.g., available CAD models). Occlusion avoidance [103] and illumination maximization [66, 104] have been considered where the model of the task environment is available. Camera placement for a known object model has been optimized by feature considerations, e.g., visibility, in focus, containment [54, 114, 115]. Some simplistic geometry-based approaches without considering surface textures have also been proposed for internal modeling [58], and most informative view assessment [27, 55]. Although these methods provide hints of what could be the best viewpoint for an object, these methods are not practical since they do not consider the surface texture and require a detailed geometric description.

4 System approach and requirements

In this section, we describe our system requirements and approach for a multi-camera DBT. We study real-time camera view management for instructional DBT scenarios with remote viewers. In these scenarios, a trainer is situated on-site with the training equipment while viewers watch the training from a remote location. We will address how the best view from all camera views can be selected, mainly based on factors derived from related work.

4.1 Camera placement

Camera placement ideally is optimized for visibility by considering trainer movement, object manipulation, performed actions, scene geometric features, and the illumination of the scene [18, 53, 68, 90]. It is desirable that at least one of the cameras should be able to cover the trainer's action for the task at hand, e.g., local, spatial, or body-coordinated [100]. It should be mentioned that, initially, we assume the cameras in the training scene to be static. Other scenarios where a camera can be re-positioned dynamically to generate a better view can

be envisioned by including a camera on a robot arm or worn on the head of the trainer, yet these are not considered in the frame of this paper. Also, it should be mentioned that we limit our requirement analysis to a single trainer-based scenario and not to multiple trainer-based collaborative ones.

4.2 View analysis and selection metrics

A view recommender system should analyze the different camera views and show appropriate content in the same view or in different views simultaneously [41]. Metrics for best view selection can be derived from literature, and inherently rely on a close-to-optimal placement of cameras. For the best view, firstly, the overall scene should be clearly visible and well illuminated (A), to afford an overall understanding of the scene, including the location (and change thereof) of objects over time. Secondly, the trainer's pose, defined by the configuration of body parts, should be visible or clearly understood during the instruction, especially those body parts relevant to the instruction (hand, arms, and potentially upper body) and communication (face) (B). With respect to the visibility of objects relevant to the training exercise, objects that are being interacted with during the instruction should be clearly visible (C). Finally, the combination of (B) and (C) results in the visibility of specific actions (D), being the movement of body parts and associated objects during an instruction. The communication view basically considers the visibility of the face of the instructor. Ideally, the full face is visible while facing toward the camera. The best view can be extracted from the available camera views for every timestep. However, the best view might change frequently, which could fail to convey any meaningful information. Moreover, measuring the best views on every consecutive timestep might not be possible due to limited computational resources. Hence, there should be a maximum frequency at which the views should change, an issue we focus on in our user study.

4.3 View strategy

The main objective of the view selection is to provide remote viewers with the optimal camera view from the set of views to maximize learning by observation. However, a single view may not necessarily capture the necessary information related to the actions and communication of the instructor. During instruction, both aspects can be separated (e.g., the instructor performing an action without speaking) and integrated (e.g., speaking while operating on an object). While vocal communication will be heard at all times, irrespective of a view, some actions during performing an action or communication may not be captured by a single view in an optimal manner. For example, a close-up may be the most appropriate view to show an action, limiting the view of the instructor's face. As such, both types of views could be considered separately to allow for higher flexibility in observing instructions. For the discussion that follows it is important to keep in mind that during an "action", an actual operation is performed on an object, whereas during "communication" this is not necessarily the case.

We refer to the view associated with showing the trainer's action as the **action view**. The action view should clearly show the trainer's body in relation to objects that are being operated, and other relevant scene objects that provide context for that action. For example, in a welding operation, the parts are to be joined together and the torch should be visible to understand the operation of the torch in relation to the object it is operated on. This view should show object geometry optimally, with minimum occlusion, and with proper illumination [66, 103, 104].

During training, the trainer will likely provide vocal instructions to aid the viewers' understanding. The vocal instructions tell the viewers about how to perform the training actions or give other relevant information. The trainer can give vocal instructions when not performing an action. They can be related to gestures the trainer makes to establish the context of subsequent actions that are going to be made [36, 75]. It may include pointing to explain the relevant action or representational gestures for communication grounding before the action is performed. To make the transfer of instructions to the remote viewer, we call the view with the trainer's face, but also part of the body, and hands that make gestures the **communication view**. This view will also be important during direct communication between the instructor and observers. On the other hand, vocal instructions can also be given while performing an action. The trainer's face and eye gaze are important for these situations as they inform the viewer about a specific object or action of interest. Note that the view may hold similar content as the action view. The difference is that the communication view will always include the trainer's face and hands for gestures, whereas this is not necessarily the case with the action view, which mainly considers the action itself.

5 System

In this section, we describe the scenario and system by which our view recommendation is achieved and evaluated.

5.1 Scenario simulation and camera placement

Our system has been developed and tested with both real-world and simulated footage. To assess our system in controlled, comparable conditions between all participants in our user study, we created a simulated environment tailored for the study. The simulation deployed a diesel engine assembly operation. Since the diesel engine parts are too big to fit on one table, they are spread over three tables. The trainer collects the parts from these tables and performs a series of assembly operations, including placing and carrying objects, and using tools. Thereby, the trainer performs all three types of tasks: local, spatial, and body-coordinated as suggested in [100], which cover all types of tasks in a DBT. Hence, our simulation setup can be regarded as representative of DBT. The trainer also describes his current actions using vocal instructions. In due course, we set the object and camera positions, as well as scene properties, e.g., illumination, and texture. Fundamentally, the images coming from the simulation have the same relevant properties as real-world imagery that are required as input to the system, even though the scene is less realistic. Eight cameras were strategically placed, their locations being optimized by pilot testing with 4 users. While our initial placement was based on our metrics, the users were not bound by these metrics to avoid potential biases. As can be seen in Fig. 1, as a result, some cameras are set for capturing overall actions on the table, some are placed closer to the objects for closeups, and a single camera is placed as an overview camera (rightmost on the second row).

5.2 High-level architecture

Based on the analysis presented in Section 4, we developed the view recommender system. Its components are depicted in Fig. 2. The recommender system will feed a web front-end similar to a conference call tool to show the best action and communication view. As this

Fig. 1 Eight cameras are placed in the virtual diesel engine assembly scene. Camera placement is shown in the lower image. Views from the eight cameras are shown in smaller images on top

front end is not the focus of this paper, it is not handled further here. Multiple camera views, tracking information of predetermined regions of interest, and scene audio provide input to the system. The view recommender should provide the best view stream of the DBT session in real-time in a coherent manner as output. The camera views are acquired from multiple cameras strategically placed in the scene (see Fig. 1). Predetermined regions of interest and some scene objects' surfaces that are important for training are usually decided by the trainer or someone with domain knowledge of the training scenario. These regions – the regions of interest we will discuss in Section 5.4 – are usually fed by a tracking system into the view recommender as tracking information. In our case, we grab the predetermined regions directly from the simulation. Since we are dealing with 2D images, this information is delivered as

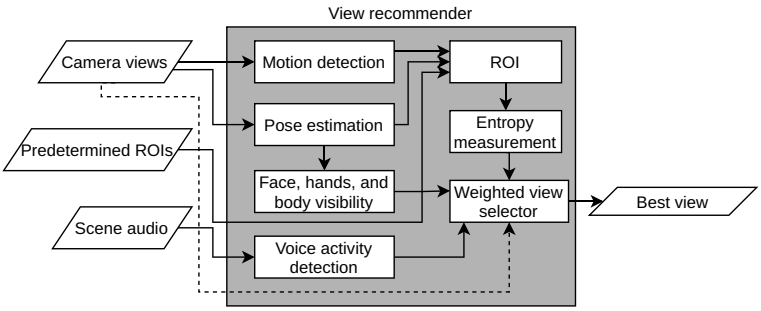


Fig. 2 Flow diagram of best view recommender system

2D image coordinates. Finally, scene audio for the trainer's voice activity detection is also sent as input. Voice activity detection only tells us if there was a presence or absence of human speech in the audio.

To detect activities, the system requires a motion detector, a human pose estimator, and a voice activity detector. The motion detector detects the region of motion in the input views while the pose estimator detects the trainer's pose by separating the image regions of the trainer's head, body, and hands from the input video stream (see Section 5.4). It detects the region of interest (ROI) for a single camera view from the union of scene motion, the trainer's body pose, and the predetermined regions. To find the amount of information in the ROI, it is fed into the entropy measurement unit. For the best view selection, the entropy measure of the ROI, the trainer's face, hands, body visibility from each of the camera views, and voice activity detection are taken into account to calculate a weighted score (shown in Section 5.3). As calculations are compute-intensive and computational resources are usually limited, different modules of the view recommender (Fig. 2) update at different frequencies. For example, both the motion detection and pose estimation modules (See next section) run at 3.6 Hz at our workstation (a thread-ripper with 128 threads). However, the weighted view selector module runs at a higher frequency that is at the frames per second of the input camera streams. This is done so that a smooth best view stream is delivered to the viewer. Since we are running the same computation steps for all of the view streams, we try to make the process as lightweight as possible. For example, to process the regions, instead of using detailed contours of detected motion and body poses to delineate detected motion and pose regions in the image, we prefer bounding boxes to improve computation performance. Moreover, bounding boxes include some of the non-detected pixels that would give some context about how the detected pixels are related to neighboring pixels, thereby facilitating better understanding by the viewer [83]. In the next sections, we will look more closely at the different sub-components of the system.

5.3 Camera view score calculation and selection

To find out the score associated with a camera view, the entropy, face visibility, hands visibility, voice activity, and predefined ROI visibility are multiplied with associated weights, and their weighted average is taken. This score measurement is shown in Equation 1.

$$score = \frac{1}{\sum_{i=0}^{5} \alpha_i} \Big(\alpha_0 H + \alpha_1 (p_{body}) + \alpha_2 p_{face} + \alpha_3 (p_{face})(p_{voice}) + \alpha_4 (p_{hands}) + \alpha_5 (p_{pROIs})(p_{body}) \Big) \tag{1}$$

Here, the weighing parameters are $\alpha_i \in \{0, 1\}, i = 0, ..., 5$. We will describe in our user study (Section 6) how these values will be compared with the ones found in our study. The scores from all the camera views are compared, where the maximum valued camera stream is our best view. In case two or more cameras have equal scores, one should be chosen randomly. From Equation 2 (described in detail in Section 5.4) and Equation 1, it can be seen that our best view selection takes into consideration the following: information content in the ROI of scene motion, trainer's body pose, and predetermined regions. It also considers the trainer's face and hands visibility, face visibility while giving instructions, and predetermined ROI visibility when the trainer is detected within the view. The information content of an ROI is higher with more visual features from scene geometry, larger features, and better illumination

conditions. To capture gestures made by the trainer, the visibility of the face is important when vocal instructions are provided. Hence, in Equation 1, p_{face} and p_{voice} are multiplied. Likewise, predetermined ROIs are only important for a camera view if the trainer is visible in that view. Hence, p_{pROIs} is multiplied by p_{body} in Equation 1. The weighted metric values are divided by the sum of the weights ($\sum_{i=0}^{5} \alpha_i$) so that the score is a weighted average of all the objective metrics. In the next sections, we will have a closer look at the ROI, image analysis, and entropy calculation.

The weighted view selector determines a score for each of the camera views. After considering spatiotemporal coherency it outputs the best view in real-time. It uses Equation 1 (see Section 5.3) to calculate the score of each of the camera views. It selects the camera view with the maximum score as the best view at that instant. However, if the weighted view selector always selected the best view based on the score at that instant, the view might change too frequently to be meaningful to the viewer. To avoid frequent switching between views, the view selector needs to maintain spatiotemporal coherency for the viewer. It should only switch the view if for some predefined time (consecutive number of frames) a view gets selected as the best view. To do so, the system uses a low-pass filter to reduce frequent view switching. It uses a window of fixed duration (T_w). We choose 2 seconds, which contains all the previously selected camera view IDs with maximum value scores. For the best view, the most frequent camera ID within this duration is selected. This way the view selector selects a camera view that is scoring maximum in the last T_w seconds. The trade-off of this low pass filtering operation is that it potentially causes a delay of $\frac{T_w}{2}$ seconds for the best view switching. The choice of this T_w is dependent on the training scenario. If a DBT has a lot of densely packed actions then this can be a small value, otherwise, it can be set to a bigger value.

5.4 Region of interest

To analyze the different views, the system identifies an ROI. An ROI is a bounding box in a camera image that is fed into the entropy measurement component. We denote the ROI of detected motion as ROI_{motion}, and the estimated pose region as ROI_{pose}. Prior to the training, some regions or parts of the training objects that are important, e.g., a surface or front side of a tool or machine, can be specified as predetermined ROI (ROI_{pdet}). The trainer or expert specifies a rectangle (or, a polygon) with a surface normal specifying the inverse of its preferred view direction in 3D space. When looking at the rectangle from the side with a surface normal and in its opposite direction, it can be clearly seen. In a real-world scene, it can for example be marked by four markers on the vertices of the rectangular region to afford marker tracking. This way the trainer or expert can incorporate their domain knowledge in the best view selection. The union of ROI_{pdet}, ROI_{motion}, and ROI_{pose} is the region of interest (ROI) view for the current frame, shown in Equation 2.

$$ROI = ROI_{pdet} \cup ROI_{motion} \cup ROI_{pose} \tag{2}$$

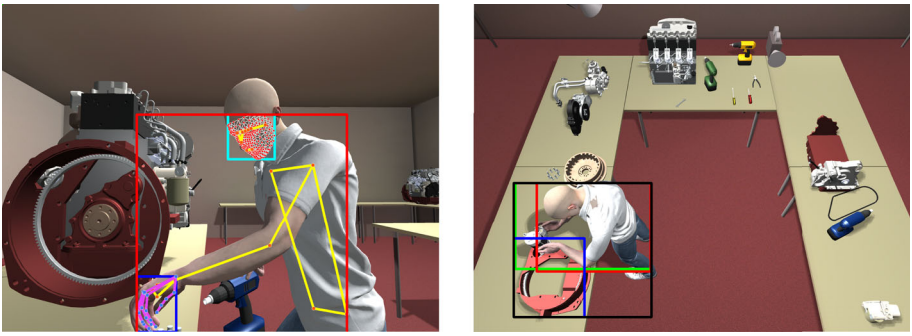
The ROI includes the pixels associated with predetermined surfaces that are vital during training. It also includes detected motion regions, i.e., all moving objects' parts and the trainer's moving limbs and body pose regions like the face, hands, arms, torso, and legs. We use entropy (H) to measure information content in this ROI (see Section 5.7). Moreover, the pose estimation module also provides information regarding face and hands visibility. A voice activity detector is used to detect if the trainer is giving any instructions

vocally. The body pose visibility (p_{body}), face visibility (p_{face}), hands visibility (p_{hands}), predetermined ROIs visibility (p_{pROIs}), and voice activity (p_{voice}) are probability values, i.e., $p_{body}, p_{face}, p_{hands}, p_{pROIs}, p_{voice} \in [0, 1]$. These values can also be Boolean values ($\{0, 1\}$) depending on the detector output.

In Fig. 3, two example images of motion detection, pose estimation, and ROI determination is shown. A bounding box is used to encapsulate the detected regions. Since we are interested in specific parts of the body skeleton of the trainer, those regions are encapsulated in separate bounding boxes. In Fig. 3a, the skeleton from pose estimation is shown. The face points are inside a light blue bounding box, the hands are in dark blue bounding box. The overall body pose is encapsulated to be used as ROI_{body} in Equation 2, shown in a red bounding box in Fig. 3. All the ROIs are shown in Fig. 3b. The estimated pose region is shown in a red bounding box. The detected motion region is shown by a green bounding box. Predetermined regions fed by a tracking system as image coordinates are shown in a blue bounding box. The union of all the ROIs, as in Equation 2, is shown in a black bounding box. The detection of the ROIs is important for our approach. Sometimes due to the partial view of the trainer, the pose estimation can be noisy. Due to the movement of shadows, movement detection can also include false positives. Hence, depending on the performance of the pose estimation or motion detection modules, sometimes the detected ROIs can be noisy. In those cases, it is recommended to fine-tune the parameters of the pose estimation or motion detection modules to increase their accuracy.

5.5 Image analysis

Image analysis is performed for motion and pose estimation to support ROI extraction. These operations are performed for all of the video streams. For motion detection, we have used



(a) An example of the trainer's body pose estimation and pose-related ROI. The Face and hands ROIs are also shown.

(b) An example of the overall ROI and including ROIs. The overall ROI is used for the entropy measure.

Fig. 3 Union of scene motion, trainer's pose, and predetermined ROIs results in ROI. Scene motion is detected along with pose estimation in the left image. Detected face and hands are shown in blue bounding boxes. By combining these regions with predetermined ROIs, the overall ROI is determined. In the right image, the pose and ROI are shown in red, the predetermined ROI in blue, the motion in green, and the encapsulating overall ROI is shown in black

OpenCV's [1] motion detection (GSOC) implementation [34]. We create an instance of a background subtractor that first learns the background from the input video stream. After that, when a new frame is fed into it, it separates the foreground. The background model is updated continuously depending on the previous frames. The foreground is encapsulated into a bounding box, which is the motion-detected ROI. For pose estimation, we have used Google's Mediapipe library [24] [2]. It uses the BlazePose [44] convolutional neural network for human pose estimation. It detects 33 body keypoints from an input image on demand. Subsequently, it estimates these keypoints by predicting these landmarks and segmentation masks, which makes it ideal for videos with low computation overhead. We need to run multiple (8) instances of this pose estimator simultaneously. Hence, we have chosen it for the real-time performance version. After the 33 body landmarks are detected by the BlazePose model, face and hand regions are predicted. From the hand, face predicted regions, face, and hand landmarks are extracted, which results in a total of 543 landmarks (33 pose, 468 face, 21 x 2 hand landmarks). For predetermined ROI, we do not perform any image-based analysis. It is given as an input bounding box (a set of 2D image coordinates) from the known simulation environment to the view recommender.

5.6 Voice activity detection

For our simulation environment, it was known when the trainer was giving vocal instructions. As such, it was used as a Boolean input to the weighted view selector. For a real-world scenario, described in Section 6.6, we have used voice activity detection (VAD) [38, 135]. A sliding window of 2 seconds is used to collect frames with PCM (Pulse-code modulation) data from 48KHz audio input. An instance of VAD, which is part of WebRTC [3] module detects if there was voice activity in that audio segment. This Boolean value was next fed to the weighted view selector.

5.7 Entropy calculations

Entropy is often used as an information measure for the content of images [109]. Entropy is expressed by the Equation 3. For a discrete random variable X with possible outcomes in the set $a_1, a_2, ..., a_n$, entropy is defined as

$$H(x) = -\sum_{i=1}^{n} p_i \log(p_i) \tag{3}$$

where, probability of a_i is $p_i = P[X = a_i]$. For continuity, $0 \log(0) = 0$ is assumed. In other words, entropy gives an average of information content for every possible outcome. Using this formula (3), entropy can be measured from image pixels' intensity values. However, this formula disregards the spatial relationship of pixels. This formula for entropy calculation will provide the same result in case the image pixels are shuffled. Since the pixel-wise shuffled image is a different image, the spatial relationship of pixels in the entropy measure needs to be accommodated.

In previous work, while calculating viewpoint entropy the spatial relationship of pixels has been incorporated to some extent by considering entropy based on visible object's faces

[1] https://opencv.org/

[2] https://github.com/google/mediapipe.git

[3] https://webrtc.org/

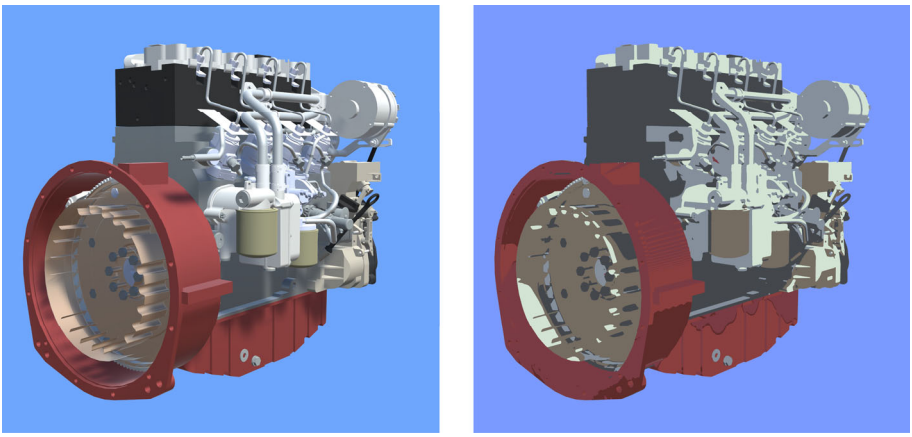
[37, 50, 126, 127, 129, 134]. This was done by grouping neighboring pixels on visible object faces. However, by doing so this viewpoint entropy disregarded the texture information. Moreover, these approaches needed a 3D model of the scene in question. For a scene S, a viewpoint p was placed in the center of a sphere and defined viewpoint entropy I by:

$$I(S, p) = -\sum_{i=0}^{N_f} \frac{A_i}{A_t} \log \frac{A_i}{A_t} \tag{4}$$

Here, N_f is the total number of visible faces, A_i is the projection of face i on the sphere, A_t is the total area of a sphere, and A_0 is the projection area of the background. Adding background projection A_0 enables normalization.

In this section, for ease of describing the entropy calculation, we assume the ROI encompasses the whole input image. Hence, we would not mention ROI separately; we would refer to the ROI as the input image. In a scene, pixels corresponding to the same surface have similar values when we disregard the high-frequency texture information. We consider those pixels as a single group of pixels or a superpixel. Superpixels can be defined as similar-valued pixel clusters that retain an object's shape. In other words, a whole image is a collection of tessellated superpixels. In Fig. 4b, each of those blobs is considered a superpixel. These superpixels convey the geometric structural nature of the scene to a viewer and discard high-frequency texture information. Our entropy formula is adopted from 4 by replacing objects' faces areas with superpixels' areas. If we consider an image to be a collection of superpixels where all the information it conveys is enclosed inside the image, we can define an area-based entropy H_a as from Equation 3 and 4:

$$H_a = -\sum_{i=0}^{N} p(S_i) \log(p(S_i)) = -\sum_{i=0}^{N} \frac{A_{S_i}}{A_{image}} \log(\frac{A_{S_i}}{A_{image}}) \tag{5}$$



(a) An example of an input camera view.

(b) Color segmentation of the input example view.

Fig. 4 An example of an input image and its color segmentation. Although after color segmentation, some high-frequency texture details are lost, it still retains the object surfaces' shape

Where $p(S_i)$ is the probability of finding a superpixel S_i in the image. We can measure $p(S_i)$ by comparing S_i's area with the area of the image. Here, the area of a superpixel or image means how many pixels it has. A_{S_i} is the area of a superpixel S_i, A_{image} is the area of the image. Since, the whole image is a collection of N number of superpixels, $A_{image} = \sum_{i=0}^{N} A_{S_i}$. The flow diagram (Fig. 5) shows the specific steps that are taken for view evaluation.

In a natural scene, extracted superpixels can vary in size and shape. For the high-frequency components removal (A) we mainly focused on the geometric features, e.g., edges, shapes, or contours of objects in the scene. Since the viewer has to recognize and track the objects used for training, we assume that due to motion blur, the shape of the objects is more important than high-frequency textures. We filter out the high-frequency components of the input images based on the assumptions for DBT that scene geometrical features and color information are prominent or most visible, and texture information is not prominent during action recognition.

A median filter is used to blur the input image to filter out the high-frequency components (A). After performing this low-pass filtering on the image, a color-based segmentation (B) is performed by clustering the pixels. We perform color-based segmentation because it assumes similarly colored pixels correspond to the same clusters. It thus conserves the shape of different surfaces in the image compared to other techniques, e.g., semantic segmentation. As low-pass filtering is done prior to segmentation, the output is a posterized representation of the input image. In Fig. 4b, segmented parts are shown by centroid colors for the input image (Fig. 4a). These segments are used as superpixels. K-means clustering is used for color segmentation. To save execution time, a fixed number of clusters is used. Depending on the scene and required accuracy, 6-15 number of clusters are selected. By definition, in a superpixel, all of the pixels have the same color. To calculate the probability of a superpixel (C), the number of pixels inside the superpixel is taken as its area and divided by the number of the overall pixels in an image. At this point, every superpixel in our segmented image is assumed to be unique in shape. Hence, they are considered separate symbols. Using the Formula 5, the primary entropy measure H_a of the image is calculated (D).

Next, the entropy for similarity measures among superpixels is updated. If there are repetitive, mirrored, or similar components in an image, its primary entropy, H_a should be reduced. It should be done because redundant components inside the image reduce the information content or complexity. Hence, the image entropy should be reduced. It should be noted that only similarities on the same scale or size level are relevant. If two regions in an image have a similar shape but differ in size they are considered dissimilar. For faster computing, grayscaling, and downsampling the input image is done prior to similarity extraction (E). As we look for similarities of shapes at equal sizes, only intensity levels are important. Hence, a grayscaled image is used for this step. Similarities can be present at different scales. Since there is no prior information about the image contents, it is needed to determine the similarity

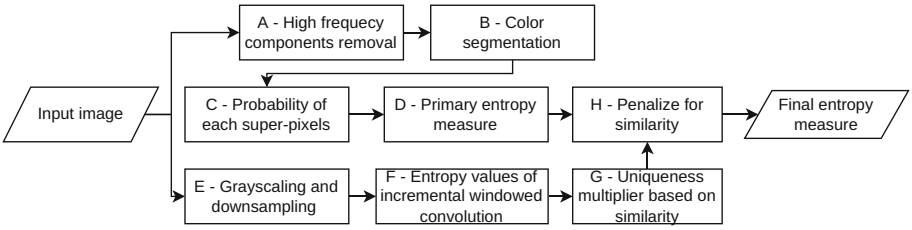


Fig. 5 Steps for incremental convolution-based uniqueness multiplier determination for entropy reduction

measures in different scales. To make this operation faster, a very low resolution (e.g., the highest image width or height is 20 pixels) downsampled image is used.

To find the degree of similarities an incremental windowed convolution is performed. The low-resolution image is convolved with an incremental normalized averaging kernel (F). When an averaging kernel or box-blur kernel is applied, it replaces each pixel with the average value of neighboring pixels. If an image has similar regions of the size of the averaging kernel, these would produce similar values. Hence, an averaging kernel of size k is used to convolve over a 2D image. If this image has similar regions of width k, it would produce similarly valued intensity peaks in the convolution output. As a result, a classical entropy measure, as in Equation 3 of this 2D convolved output will produce less entropy compared to the image before convolution. We use an incremental kernel of size, $k = 3, 5, ..., \lfloor \frac{max(width_{image}, height_{image})}{2} \rfloor$. We use odd kernel size because the input grayscale values are spread within an equal-length span around the output pixel. The size of the kernel starts from a minimum possible length of 3 until the maximum possible length which is half of its shorter side length.

$$K = \frac{1}{k^2} \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix}_{kxk} \tag{6}$$

Equation 6, shows the formula for box kernel. For a low-resolution image of a maximum side (width or height) of 20 pixels, k = 3, 5, 7, 9. Using the classical equation of entropy as in Equation 3, we calculate the entropy profile for the convolution output, H_k. In Fig. 6, each row shows the convolution output for the incremental kernel sizes of 3, 5, 7, and 9 for some test images. If we observe carefully, for 'machine's collection', in the third row, $H_{k=9}$ is the lowest entropy. This happens because at $k = 9$ size, the low-resolution image has similarity present in it.

If multiple kernel sizes are used, the least entropy of the convolved output will be found when the kernel size is equal to the similarity size. Likewise, the most entropy will be found when the kernel size fails to capture any similarity. Hence, we can use these two kernel sizes to find the reduction factor by which the primary entropy H_a of the image should be reduced.

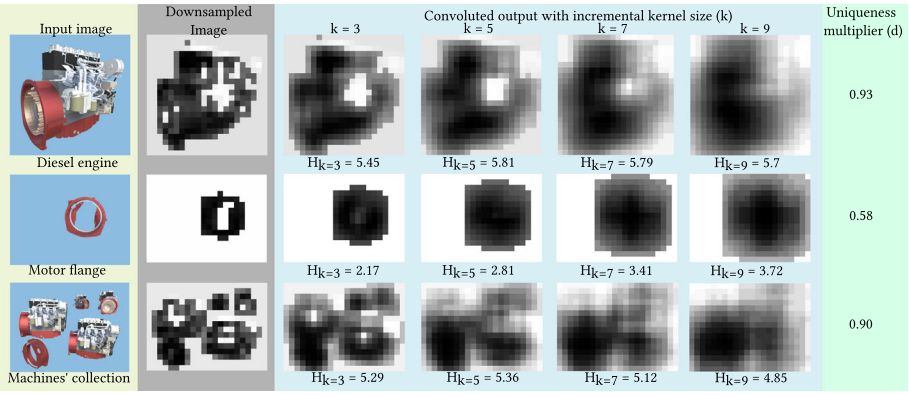


Fig. 6 Shows the convoluted output with incremental kernel size, $k = 3, 5, 7, 9$ on low resolution grayscaled image. From the convolutions, the uniqueness multiplier is determined. For better depiction, the convolutions have been shown after histogram equalization

We call it the uniqueness multiplier of the image. We define the uniqueness multiplier as the ratio of the maximum similarity kernel size to the minimum similarity kernel size (G). The uniqueness multiplier (d, where $0 \leq d \leq 1$) is expressed by Equation 7.

$$d = \frac{min(H_k)}{max(H_k)} \tag{7}$$

In Fig. 6, we found that the motor flange has a maximum uniqueness multiplier compared to the diesel engine and machine's collection. The motor flange is circular in shape, i.e., it is the most similar region. As a result, it has the maximum uniqueness multiplier value.

By multiplying the primary entropy H_a with this uniqueness multiplier the final entropy value (H_{ROI}) is determined (H). It is shown in Equation 8.

$$H_{ROI} = H_a * d \tag{8}$$

Finally, until now, we have described how to determine the entropy of an image. But in the view recommendation, the entropy of the whole image is not of interest, but rather the entropy of the ROI. Hence, we denote the image entropy value as H_{ROI}. However, the ROI is a subset of the view. Hence, to determine the entropy of the view, we want to include a scaling factor for the entropy. A scaling factor is included by multiplying the entropy of the ROI with the ratio of the ROI area to the whole view area, shown in Equation 9.

$$H_{view} = H_{ROI} * \frac{A_{ROI}}{A_{view}} \tag{9}$$

In Equation 9, A_{ROI} is area of ROI and A_{view} is area of the camera view, $A_{ROI} \leq A_{view}$.

6 User study

In this section, we report on a crowd-sourced user study to compare manually selected views with the ones from our automated selection method and get insights on the metrics that contribute to the best view selection.

6.1 Goals

We compared the user-selected best view choice with the automated recommendation view by our system. In our simulation environment, there were eight views from eight cameras. In a first pass, the results (rank list from ratings) were compared with the best views selected by our system to see to what extent they match. Furthermore, to address the definition of what constitutes a "best view" study participants were asked to rate our initial set of metrics as well as provide indications for further aspects that can be analyzed. Based on these statements, the weights of our metrics were adjusted, and an additional comparison was performed to address potential performance improvements.

6.2 Procedure and design

The user study was designed as an online experiment in which users watched eight video streams and selected the best view at predefined intervals. Participants only rated action views, i.e., the views that show the trainer's actions in the scene. Based on the scenario described in Section 5.1, we produced a video in which all eight generated, equal-sized views would be

seen at once (similar to the top two rows in Fig. 1). Each view was sized equally in the video which had a resolution of 1920x1080 (Full HD). We produced a single tiled view of all eight camera views, instead of recording eight views, to avoid potential synchronization issues. The placement of the cameras was fine-tuned in a pilot study with 4 participants who critically observed the camera footage. This placement was not bound by the metrics we selected as part of our recommendation system: users were free to make suggestions based on any kind of criteria. Audio was included in the video, with aural instructions of the different steps in the montage scenario. Participants selected the best view every five seconds, resulting in 59 measurement points in our video (4:55 minutes).

The best view selection was a three-step process: users first selected the best view from the eight camera views, then rated their confidence in the selection (7-point Likert scale, 7 being the best), followed finally by an initial assessment of the quality of each camera view, again on a 7-Point Likert scale. The video would only continue after all cameras were rated. We did not instruct the user to base their judgment of the best view on any specific criteria: instructions were provided prior to the experiment that only included the general goal of selecting the best view on the basis of best understanding the instruction by the trainer. As a result, users selected the best view that depicted the trainer's actions optimally. After finishing all 59 trials, a final questionnaire appeared. This questionnaire included demography questions and a set of specific questions about best view selection. The view selection-related questions were divided into several main categories: overall scene geometry visibility and scene illumination, visibility of objects and body parts relevant to the instructions, visibility of predetermined regions, and the visibility of specific actions. These categories are in line with how we determine the best view using our metrics. To allow users to suggest further metrics, we included open questions at the end that specifically asked for other aspects that users felt contributed to their selection of the best view. The online experiment was performed on a website hosted by one of the authors. We used Prolific as a crowd-sourcing host service to source the experiment to participants worldwide. On average, the experiment took around 37 minutes to finish. Participants were awarded around 5 Euros for their participation, calculated along with best-practice guidelines provided by Prolific.

6.3 Results

A total of 43 participants (age 18-40, 20 female, 22 male, 1 non-binary) took part in the study. We analyzed the aggregated data from all of the participants by summarizing and comparing it with the recommender system-generated view selection. A total of 20,296 ratings were fed in the analysis, produced by the 43 participants at 59 intervals for 8 cameras. All of the participants had also mentioned a best view camera ID for all timesteps, thus generating additional 2537 datapoints, which were also used in the data analysis. The dataset includes the best view camera ID, and view rating for all of the camera IDs collected every 5 seconds. We have summarized the data from all participants into a single summary dataset. The summary of best view camera IDs was selected by taking the most frequently selected one at each timestep. The summary of all the individual camera view ratings at each timestep was created by taking the median of those ratings. This was done to avoid outliers in the measure of central tendency. From our automated view recommender system, we also have the best view selection and scores for all the camera views. These scores from Equation 1 can be interpreted as ratings/relevance of the camera views at each timestep. We call them **view-recommender ratings**. To evaluate the view-recommender system, we would apply different techniques to compare and correlate user ratings with view-recommender ratings. However,

in Equation 1, initially the coefficients (α_i) are unknown. Hence, initially, we assume all the α_i values are 1. Based on the insights gained from the questionnaire, we updated these values subsequently.

The summary user ratings are compared with view-recommender ratings for each timestep. We do so by using Normalized Discounted Cumulative Gain (NDCG). NDCG is a measure of ranking quality. NDCG is the ratio of the Discounted Cumulative Gain (DCG) [51, 79] of recommended order to the DCG of ideal order ($IDCG$). The following formula illustrates this principle:

$$NDCG_p = \frac{DCG_p}{IDCG_p} = (\sum_{i=0}^{p} \frac{rel_i}{\log_2(i+1)}) / (\sum_{i=1}^{|REL_p|} \frac{2^{rel_i}}{\log_2(i+1)}) \tag{10}$$

In Equation 10, $NDCG_p$ is the normalized DCG accumulated at rank position p, REL_p is an ordered list of items according to their monotonically decreasing relevance until rank position p, from which Ideal DCG ($IDCG_p$) is measured. In our experiment, we consider NDCG for all of the eight camera ranking at any timestep, i.e., $p = 8$. If we consider data from each of the timestep as a sample, then the overall NDCG score is M=0.906, med=0.92, SD=0.07, where $0 \leq NDCG \leq 1$. This high NDCG score shows that even with no adjustments of the coefficients (α_i) values in Equation 1, the view recommendation performs very well as it reaches 90.6% effectiveness of the best possible ranking. We are unaware of any other work using NDCG for describing the performance of view recommendation, hence, we could not compare it against a baseline. In Fig. 7, the NDCG score for the duration of the whole experiment is shown. It shows that the NDCG score is mostly high with some outliers in the last quarter of the simulation video. We found that the topmost ranked camera ID by the view-recommender matches the topmost ranked camera ID from the summary of user rating 42.4%, the 2nd most ranked 11.9%, and the 3rd most ranked 10.2% times.

We show the percent of different camera views chosen as the best view and how the view-recommender performs before and after optimization (of coefficients in Equation 1) compared to the summary or representative ratings by the users in Fig. 8. In that figure, we show all of the user's ranked views, and the topmost view-recommender ranked camera

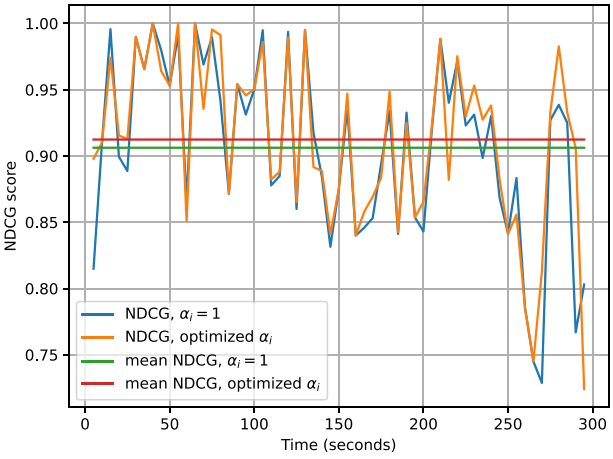


Fig. 7 Optimization of weights of metrics shows a slight improvement in NDCG values. NDCG values for all of the timesteps, when $\alpha_i = 1$ and with optimization of the alpha values (see Section 6.4)

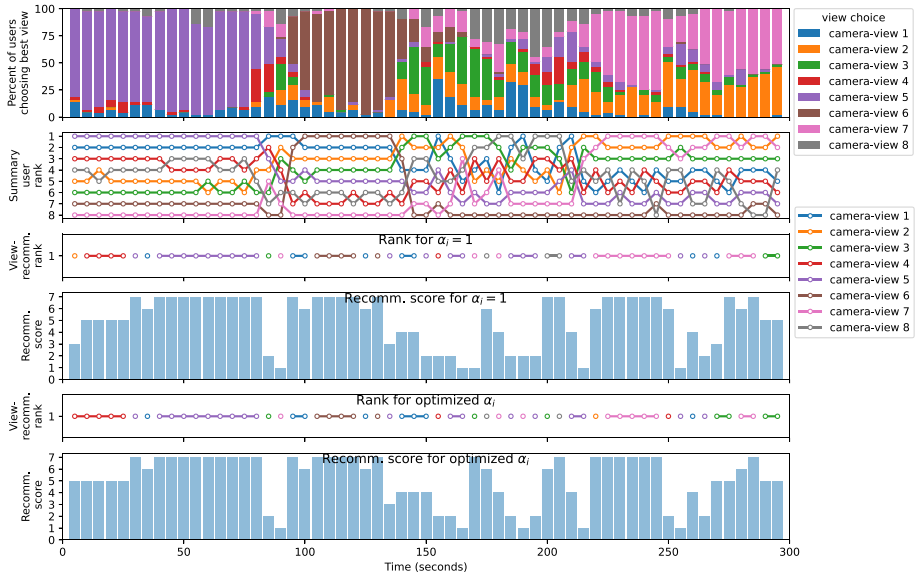


Fig. 8 View choice percent made by participants (n=43)

IDs over time. We also show a custom recommendation score over the video timeline. This custom recommendation score was given 7.0 if the topmost ranked view-recommender ranked camera ID matched the topmost, 6.0 if it matched 2nd ranked, 5.0 if it matched 3rd ranked in summary user ranking, and so on. Before optimization, described in Section 6.4, this recommendation score was found to be: M=5.03, med=6.0, SD=2.16 (see 4th sub-figure in Fig. 8), and with optimization it improves: M=5.11, med=6.0, SD=2.09 (see 6th sub-figure in Fig. 8). The topmost sub-figure in Fig. 8, shows the percent of different camera views chosen as the best view at every timestep. Although there has been a general consensus about the best view in the first 1.3 minutes of the videos, the users have chosen different views in the middle part of the training, and in the last 1.4 minutes, there was basically a bi-modal distribution for the view choice. The summary user rank does not capture this, because it is obtained from the view rating of individual camera views. From the recommendation score in Fig. 8, it is clear the view-recommender has worked well when there was a general consensus about the best view among the participants. It also performed well when basically two views were almost equally prominent. However, when the participants could not reach a consensus about the best view, the recommendation score was worse. From those recommendation scores, we can say that the recommended view is almost always one of the first three best views rated by the users (Fig. 8).

We present all the survey questions regarding view selection choices in Table 1. In Section 6.4, we choose a subset of the questions to determine the weights of the metrics for view recommendation. To better understand the contribution of each of our metrics for view selection, we calculated the NDCG scores with a single coefficient equal to 1 and all others to zero, i.e., $\alpha_i = 1, \alpha_j = 0$, where $i = \{0, ..., 5\}$, $j = \{0, .., 5\}/i$ in Equation 1. This way we can analyze how each of the metrics contributes to the best view selection. Table 2 shows that the trainer's hands visibility and information measure of ROI contributes most to the view recommendation, whereas predetermined regions in the trainer's presence contribute least. The metrics are not independent of each other, e.g., with the visibility of the trainer's

body pose, the probability of the visibility of the face is high. For this reason, in Table 2 we see that the NDCG scores do not drastically change even if only one metric is considered.

As a next step, we used the summary ranked list of user view and view-recommender ranked list at each timestep and performed Kendall's τ and Spearman's ρ correlation tests at each of the intervals to find if both of these ranked lists are correlated. The correlation is found frequently (p-value < 0.05) in the first quarter of the experiment, while for the rest of the time it is rare (Kendall's τ: M=0.311, SD=0.286), Spearman's ρ: M=0.413, SD=0.359)). Since the camera view IDs are constants rather than natural numbers, the correlation tests did not perform very well.

Because the camera IDs are constants, the best view time series is a sequence of constants. Hence, we decided to use edit distance [95] to distinguish between different best view series. The summary user-selected best view series is a sequence where each of the camera IDs at a timestep is the most frequently chosen one. Similarly, we also extracted a sequence from the best view from the view-recommender output. Between these two sequences, out of 59 timesteps, 26 times camera IDs matched. To compare these two sequences, we use the Demerau-Levenshtein distance[46]. This distance is measured over a small window of 2 to 5 timestep (the equivalent of 10 to 25 seconds) because there might be a temporal delay before the two sequences show similar values. We found a Demerau-Levenshtein distance of M=0.587, SD=0.064. This distance is representative of the outcome that 58.7% of the time the best view-recommendation did not match the summary best view by users. This improves a little when we used windows of different sizes with interval numbers of 2 to 5 to plot the Demerau-Levenshtein distance to compensate for the temporal noise in view selection.

6.4 Implementation adjustment and results

Until now, all the values were calculated with equal weight or coefficients ($\alpha_i = 1$) in Equation 1. However, it is important to know the optimal values of these weights (α_i) that maximize the view recommendation performance, i.e., maximize the NDCG score. This section reports on various approaches for optimization of the NDCG score, shown in Table 4. As a first step, we grouped a subset of the questions, mentioned in Table 1, into categories that are aligned with our determined metrics for view recommendation. We normalized the responses on a scale of 0 to 1 and average them to find the coefficients for Equation 1, shown in Table 3. This table shows there are six main factors that contribute to view selection. All of the sub-factor values are from the questionnaire with the responses normalized. First, the mean of different factors from Table 3 was used in Equation 1 to optimize the view recommendation score.

After plugging in the value of the factors, an improved NDCG score was achieved: M=0.912, med=0.923, SD=0.063 (the initial NDCG score was (0.906)). A slight improvement in the recommendation score was seen: M=5.12, med=6, SD=2.09 (see Fig. 8). We also repeated the correlation tests. With the optimized view recommendation, the correlation test results are Kendall's τ: M=0.383, med=0.429, SD=0.293), Spearman's ρ: M=0.479, med=0.5, SD=0.347). The correlations show an improvement over the previous analysis. The Demerau-Levenshtein distance with the optimized view recommendation is M=0.59, SD=0.048, showing again a slight improvement over the previous distance. Interestingly, note that from all optimizations, this simple approach actually was the most successful.

As a next step, to find the coefficients of the metrics in Equation 1, a confirmatory factor analysis (CFA) was used for the information measure of ROI, the trainer's body pose, and

Table 1 Mean and standard deviation of all the survey questions

Questions	Responses (1-7)	
	Mean (M)	Std (SD)
I have selected the best view by		
... understanding of scene geometry	5.12	1.59
... illumination of operation area	4.74	1.59
... an unobtrusive view of the operation area	6.21	1.32
I have selected the best view by the visibility of		
... the trainer's hands	5.58	1.35
... the trainer's face	3.63	1.77
... the trainer's upper-body	4.28	1.67
... the trainer's full-body	3.37	1.75
... the tools that were used when an action was performed	6.47	0.63
... the trainer's actions	6.63	0.58
... the trainer's upper-body movement while performing an action	5.09	1.32
... the trainer's full-body movement in the scene	4.26	1.43
I prefer to have in my view,		
... other objects that are relevant to the instruction given	2.07	2.00
or,		
... other objects that the instructor is operating on	4.81	2.25
I have rated low the views where I could not see		
... the hands performing some action	6.35	1.15
... the face when instruction was given	3.47	1.64
... the table that the training was performed upon	5.12	1.89
... the tool that was used for training	6.28	1.32
I could follow the instructions/explanations given by the trainer	5.67	1.34
I have selected the best view by		
... better understanding of the trainer's actions	6.35	0.87
... the anticipation of the movement of the scene elements (trainer, objects, etc.) in the next several frames	5.33	1.41
When switching between viewpoints		
... camera viewpoint differences should be kept minimum	4.86	1.60
... trainer should be included in the consecutive views	5.00	1.29
... operation area should be included in the consecutive views	5.35	1.21
... both the trainer and operation area should be included in the consecutive views	5.14	1.39
... there should be minimum or no movement in the action area i.e., it should only switch after the current action is finished	5.00	1.79

Note: All of the responses are on a Likert scale where 1 is minimum and 7 is maximum

Table 2 Contribution of each metric separately for view recommendation interpreted in terms of NDCG score

Condition	Contributing metric	NDCG		
		M	Med.	SD
$\alpha_0 = 1, \alpha_i = 0, i \neq 0$	Information measure of ROI	0.908	0.914	0.068
$\alpha_1 = 1, \alpha_i = 0, i \neq 1$	Trainer's body pose	0.838	0.838	0.066
$\alpha_2 = 1, \alpha_i = 0, i \neq 2$	Face	0.884	0.914	0.075
$\alpha_3 = 1, \alpha_i = 0, i \neq 3$	Face with voice	0.853	0.848	0.065
$\alpha_4 = 1, \alpha_i = 0, i \neq 4$	Hands visibility	0.910	0.939	0.079
$\alpha_5 = 1, \alpha_i = 0, i \neq 5$	Predetermined ROIs with trainer	0.824	0.830,	0.073

Note: M: mean, Med.: median, SD: standard deviation

Table 3 Normalized responses from the questionnaire are averaged to be used in Equation 1

Metrics	Related questions	Values	Mean
Information measure of ROI	Scene geometry	0.69	0.73
	Scene illumination	0.62	
	Unobtrusive view	0.87	
Trainer's body pose	View of upper body	0.55	0.54
	View of full body	0.40	
	View of upper body movement	0.68	
	View of full body movement	0.54	
Face	Face visibility	0.43	0.43
Face with voice	Face view when instruction given	0.41	0.41
Hands	Trainer's hands view	0.76	0.84
	Tools in action view	0.91	
	Trainer's action view	0.94	
	Hands performing actions	0.89	
	Tools view	0.88	
	Understanding of actions	0.89	
	Operated objects view	0.64	
Predetermined ROIs with trainer	Table view	0.69	0.43
	Relevant objects view	0.17	

hands. Since some of the metrics did not have at least three items, i.e., face, face with voice and predetermined ROIs with the trainer, a factor model could not be used for those because they would be under-determined. The average values for those factors found in Table 3 were used. For the other metrics: information measure of ROI, the trainer's body pose, and hands, the factor loadings from the CFA model were averaged. We found that NDCG score of M=0.91, med=0.92, SD=0.061; Demerau-Levenshtein distance of M=0.604, SD=0.055; Correlation test results: Kendall's τ M=0.256, med=0.357, SD=0.293, Spearman's ρ M=0.462, med=0.5, SD=0.347. Until now, we have used psychometric analysis to find out the values of coefficients in Equation 1. We wanted to perform numerical analysis to find out α_i that optimize view recommendation in terms of NDCG score. In other words, the optimized weights should

Table 4 Summary plot for view evaluation coefficients and performance metrics

Method	α_0	α_1	α_2	α_3	α_4	α_5	NDCG	τ	ρ	DL dist.
Equal weight	1.00	1.00	1.00	1.00	1.00	1.00	0.906	0.31	0.42	0.59
Simple average	1.00	0.75	0.60	0.57	1.16	0.60	0.912	0.38	0.48	0.59
Factor analysis	1.00	0.95	0.66	0.62	0.58	0.65	0.910	0.26	0.46	0.60
Function approx.	1.00	0.43	0.28	1.28	0.13	0.65	0.908	0.31	0.41	0.61

Note: τ, ρ are correlation coefficients, the former one is Kendall's τ and the later one is Spearman's ρ

reflect users' choices as closely as possible. Using the CFA model we found the approximately same NDCG score as in the simple average method reported in Table 4.

Since we have a summary dataset from users' selection, it can be compared with view selection with randomized coefficient (α_i) values in Equation 1 to create a set of (α_i, NDCG) pairs. This randomized α_i with resulting NDCG values can be used as input-output pairs to model the goodness of view selection. Because it is a highly non-linear process, we opted to use a neural network for this purpose. In Fig. 9, the basic mechanism is shown. The α_i values are given as input to both the view recommender and a neural network. The α_i values are used to calculate NDCG values by comparing view recommendations with people's ratings which is again compared with the neural network to produce an error signal. This error signal is used to update the neural network's weights. It was a fully connected two-layered network with 20 rectified linear unit (ReLu) activated neurons on each layer. After optimizing it with Adam [128] and a learning rate of 0.001, we obtained a validation error of 0.39. Our dataset was small with only 30 samples (we took 30 random alpha values), which was further divided into an 80% training set and a 20% test set. A better approximation could likely be achieved with a larger training set. Next, we performed a search by performing BFGS [80] optimization in the alpha space that maximizes the NDCG score. We found the following values of α_i respectively: 0.93, 0.4, 0.26, 1.19, 0.12, 0.6. For these α_i, we found that NDCG score of M=0.908, med=0.913, SD=0.06; Demerau-Levenshtein distance of M=0.605, SD=0.059; Correlation test results: Kendall's τ M=0.305, med=0.286, SD=0.29, Spearman's ρ M=0.414, med=0.476, SD=0.346. We were not able to approximate the NDCG score given the alpha values as the global optimum was not reached, as for most views users did not agree on a

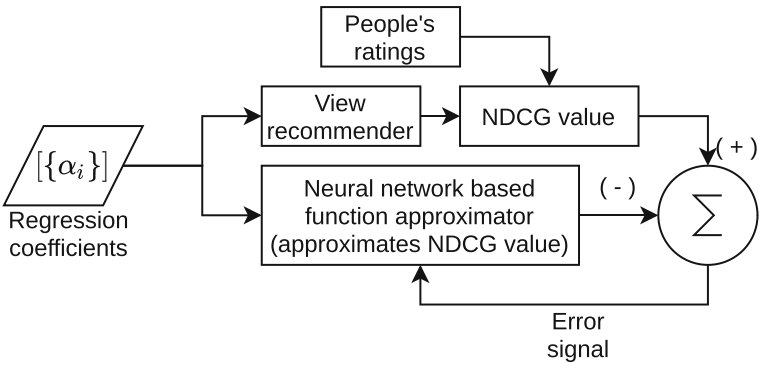


Fig. 9 A neural network has been used to model the goodness of view selection, which optimizes the NDCG value of view recommendation with user ratings

single best view (seen in Fig. 8). The performance of this neural network-based approach was comparable to the simple average method reported in Table 4.

To visualize the performance of all the reported approaches, Table 4 was included. It summarizes the different methods for determining the coefficients, i.e., α_i in view evaluation Equation 1. Since Equation 1 mainly measures a weighted average, we can multiply the α_i in such a way that their ratio holds. In this case, for a specific method, we have divided each of the α_i by the α_0, i.e., the coefficient related to information measure. Other performance metrics are also mentioned for comparison. In Table 4, we see that the simple average method has performed best, and 'Hands' was the most important metric.

As a final remark, we also asked the participants in the post-experiment questionnaire how often the view should be switched. Of all the participants, 46% voted for every 10 seconds, 23.3% for 5 seconds, 16.3% for 30 seconds, 11.6% for 3 seconds, and only 2.3% for every minute. However, our view recommender used a moving window approach, described in Section 5.2. Currently, it selects the view which has been selected the maximum number of times in the last 2 seconds. Although most people think that the view should switch every 10 seconds, sometimes the time between consecutive actions by the trainer are shorter, hence, in this case, it would probably be better to switch the view quicker.

6.5 Comparison with other implementations

Our work represents the first attempt to integrate view entropy with instructional design guidelines to determine a parameterized score for view ranking in DBT. Furthermore, we are not aware of any other studies utilizing NDCG to assess the performance of view recommendation. As a result, it was not feasible to compare our approach against a view recommendation approach for DBT. Moreover, a comparative analysis with viewpoint entropy [37] is difficult because viewpoint entropy is developed for 3D image-based rendering and it does not say how to extend it to 2D images. Although later texture handling was introduced for viewpoint entropy [126], it still requires a 3D model of the scene and does not work with 2D images. It should be mentioned that viewpoint entropy does not consider similarity, which is incorporated into our approach. Moreover, the viewpoint entropy approaches have not considered human presence, which we do for DBT scenarios. All these reasons make it difficult to compare our approach with viewpoint entropy. We compared our entropy measure with classical entropy implementation [109]. Results of this comparison are shown in Table 5.

The first condition in Table 5 represents the case where the full frame was considered as the region of interest (ROI), and only entropy was taken into account, disregarding other parameters. This is the usual condition when classical entropy is used for view selection. Out of 59 view ranking trials, 53 times the view recommender's output did not match the

Table 5 Comparison with classical entropy implementation

Condition	Entropy	NDCG	τ	ρ	DL dist.
(1) ROI=full, $\alpha_0 = 1, \alpha_i = 0, i \neq 0$	Classical	0.923	0.47	0.59	0.84
(2) Opt. ROI, $\alpha_0 = 1, \alpha_i = 0, i \neq 0$	Classical	0.914	0.46	0.57	0.57
(3) Opt. ROI, Opt. α_i	Classical	0.924	0.44	0.55	0.55
(4) Opt. ROI, $\alpha_0 = 1, \alpha_i = 0, i \neq 0$	Ours	0.906	0.31	0.42	0.59
(5) Opt. ROI, Opt. α_i	Ours	0.912	0.38	0.48	0.59

Note: Opt. means Optimal

users' choice. The first rank was a match with users' rank 13.56 times, the second rank 49.15 times, and the third rank 23.72 times. However, we found high NDCG for this condition despite the big DL distance. It was mainly because the second rank was matched quite a high number of times. In the second condition, we have introduced the optimal ROI described in our approach (see Section 5.4), as opposed to the first condition. As a result, the DL distance was reduced. Next, in the third condition, we have used the optimal α_i found by the simple average method in Table 4, which yielded the maximum NDCG scores. In the fourth and fifth conditions, we have put the results from our approach with only entropy and optimal entropy respectively. Upon comparison, it was found that the combination of classical entropy with our instructional design parameters outperformed our method (Condition 3). However, it is important to note that this approach does not consider the spatial entropy and similarity of the information content.

Fig. 10 An example of view-recommendation. The recommended view is shown below the camera views in a bigger window. Courtesy of Opdenhoff Technologie GmbH

6.6 Real-world scenario

Next to our simulation environment, we have also applied our algorithm to real-world videos. In Fig. 10, we show a snapshot of a real-world scenario. It shows 9 views from different cameras placed around a cabinet being installed. The best view is shown in the lower (large) image. Figure 11 depicts a set of views at different timesteps with the recommended view highlighted in a red rectangular boundary. The views in each of the rows are from same the timestep, often showcasing informative views. It should be mentioned that in the real-world scenario, we did not define any predetermined region. Hence, the multiplying coefficient with p_{pROIs}, i.e., α_5 in Equation 1 was zero. The view recommender still worked well.

7 Conclusion

Hereafter, we discuss the results of our approach and study. Afterward, we reflect upon our limitations and potential future work.

7.1 Contributions and reflection

Viewpoint entropy can be used well for evaluating views from 2D images in real-time. In previous work, best view selection was mostly concentrated on cases where the 3D scene or object geometry was known [15]. Some earlier work has dealt with better visibility of geometrical features [27, 55], the least amount of occlusion [103], and better illumination [66, 104]. Later on, information-theoretic approaches, e.g., viewpoint entropy [37] or KL distance based [122] were introduced. They gave a way of measuring the goodness of a view. Although viewpoint entropy gives a way to compare views, previous works [37, 94] did not address how to apply it to real-world images where the scene geometry is unknown. We took inspiration from the viewpoint entropy [37] and extended it for real-world images. Our results show that it is useful as a universal way of comparing views, since, we can compare information content in two or more views regardless of their projection types, e.g., orthographic or perspective. Unlike earlier approaches [127, 129, 134], for our approach we do not need the 3D models of the objects in a camera view, required to find the projections

Fig. 11 Several examples of recommended views are highlighted in red. Courtesy of Opdenhoff Technologie GmbH

of objects' faces. To circumvent this, we have introduced the use of posterization. We used color segmentation to find surface projections of the scene objects onto the view. Although we lose high-frequency details, e.g., texture information for doing this, it gives us a way to incorporate spatial relationship of pixels in the image, which is missing in the classical entropy definition [109]. It should be mentioned that we can control how many high-frequency components would be considered by increasing the cluster numbers for segmentation. We also incorporated a similarity measure in entropy. We penalized the entropy of the view with local and global similarities. Previous work did not focus on penalizing the view entropy measure for similarity. We found empirical evidence that it can compensate for underlying local or global similarities in a view. However, a rigorous analytical analysis should occur in future work to more closely elaborate on this. Finally, our approach is also fast and can be implemented for real-time view evaluation performance. Regarding runtime, color segmentation is the most expensive step. Since we are using k-means for color segmentation, real-time view evaluation becomes achievable.

View management in a DBT scenario requires useful metrics for analysis. Some research and commercial systems are available for live lecture capturing [2, 5, 112], instructional design guidelines are available for lecture videos [73, 84, 85, 96] and DBT [116]. However, we are unaware of previous work on view recommendations for real-time 'how-to-do' demonstration-based tutorial videos with real-world physical objects. These videos not only need to show the geometrical features and textures of the relevant scene objects clearly but also require to show the trainer performing actions on them. Although some view recommendation studies compared first person and over-the-shoulder view in DBT scenarios [32, 48, 74], these studies are limited by only considering two views. In our work, we targeted to overcome this gap by formulating suitable metrics covering scene geometric features, predetermined regions, trainer's face, hands, and body visibility with emphasis on face during vocal instructions for multi-view systems that offer a multitude of cameras. We formulated a view recommendation score, which can be used to compare view goodness. Using all of those metrics for the view selection scenario, we found out that our approach has a high NDCG score (0.912), which mimics viewers' choice well overall.

Image sub-metrics are not equally important, weighting shows slightly improved results. The score Equation 1 uses a weighted average of the metrics from a view. We have tested how these factors contribute to view visibility score when considered in isolation (see in Table 2). However, not all of these metrics might contribute equally to view selection. We experimented to find out the metrics' relative importance and also how those individually affect the views recommendation. It is not surprising to find that the viewers mostly decided the view selection solely based on the visibility of hands (NDCG=0.910, Table 2), since hands are directly contributing to actions by the trainer. Hands are also vital for communication using gestures [6, 30, 36, 106]. Hence, the visibility of the trainer's hands performing actions is of utmost importance. In our system, the ROI covers the trainer's pose, the image regions of scene motion, and predetermined ROIs. The trainer's pose conveys information through gestures and demonstration of actions. Scene motion captures the information change in the camera view which enables viewers to find out what is going on in the training scene. The predetermined ROI includes any region which the trainer had determined prior to the training. Hence, the entropy solely from ROI (NDCG=0.908, Table 2) basically measures how much information is within it. The face visibility of the trainer is also important in instructional videos [69, 72, 119, 120] aligning with the trainer's face visibility score in our study (NDCG=0.884, Table 2). Accordingly, the face is incorporated in the communication view during vocal instructions. The trainer's face also shows eye gaze during instruction, which is important during training [88]. The face visibility during instructions (NDCG=0.853, Table

2) also had a moderate score. The trainer's body visibility can influence the learning effect [87] and had a slightly lower score (NDCG=0.838, Table 2). The visibility of predetermined regions also had a good score (NDCG=0.824, Table 2). However, these regions only matter when the trainer is also visible in the camera view. When all of these metrics are assumed to have equal importance, the NDCG score was found to be 0.906 or when optimized (in Table 3) 0.912, which reflects that the determined objective metrics already reflect people's choice.

7.2 Limitations and future work

Our developed view recommender is able to mimic people's choices well. We have improved and extended the idea of viewpoint entropy to measure information content in a view. However, some limitations can be noted. First, a comparative analysis could not be performed between our entropy implementation and viewpoint entropy [37] (explained in Section 6.5). However, we have compared our entropy measure with classical entropy implementation [109] (see in Section 6.5). Secondly, the posterization by color segmentation of the input streams can be a weakness under certain conditions. We have used k-means with a predetermined number of clusters, which potentially can be an issue if the input image has a lower number of intensity clusters. In this case, our method can overestimate or underestimate the number of cluster centers, which in turn can be detrimental to the posterization process. The number of clusters has to be decided based on the scene. To improve upon this, an adaptive clustering method (e.g., k-means++ [71]) can be used without extending execution time so that the camera views could be segmented properly. Furthermore, we have considered gestures for view selection, which has translated into the inclusion of the trainer's limb visibility in the objective metrics. However, in the determined metrics, we did not consider a pointing gesture specifically, which is often highly relevant for instructions. We intend to consider the trainer's pointing gestures, and eye gaze in future iterations for better performance. Another factor we did not directly consider that may affect learning is attention. In the future, it would be interesting to extend the analysis by using eye tracking and addressing which camera view, body parts, and activities the user focuses on, and for how long. With our current analysis, this could not be addressed. In due course, further performance metrics could be introduced. For example, it will be of interest to correlate learning progress, cognitive load [93], and situation awareness measures [28] to more deeply understand the underlying perceptual and cognitive mechanisms during a training session. With respect to instructors, different training scenarios may introduce new challenges, in particular when multiple instructors – each with a different role – will require switching between instructors. Finally, although NDCG scores are high in our online study, the correlation of best-selected camera IDs with the chosen camera IDs by the study participants has some fluctuations (discussed in Section 6.3). A probable cause is the smoothing out of high-frequency camera switching for maintaining spatiotemporal coherency. To circumvent this issue we have also tried out the Demerau-Levenshtein distance between these two camera ID streams, which is also found to be high. To compensate for the delay of view switching, we also tried this distance measure with a 2-5 instance length of window size. However, none of this yielded closer distance measures.

In conclusion, in this paper, we reported on what objective metrics could be introduced for view selection in a DBT scenario. We found that information content in the ROI, comprising of scene motion, trainer's body region, and predetermined regions, is one of those metrics. To measure this information content or entropy, we have adapted and improved the idea of viewpoint entropy. Our entropy implementation improves over the classical and viewpoint

entropy by extending it from 3D models to 2D images and incorporates similarity penalization or uniqueness consideration. In DBT scenarios, our study indicated our method already achieved high accuracy in selecting the best views. Further adjustments can likely achieve even higher matching, based on the specific training scenario requirements. The degree of posterization via clustering could be adjusted for deciding the level of detail to be considered. We do not explicitly consider occlusion. If the amount of occlusion is known that could be added as a penalty factor. Having more well-defined predetermined regions could improve the determination of regions of interest. Currently, our approach only considers the face view of the trainer, which could be improved if a directly communicating face view is considered. Finally, modeling the view recommendation with more participants would reduce the noise which could lead to better performance.

Acknowledgements This work was partly funded through the Campus to World project (Innovative Hochschule, BMBF, FKZ: 03IHS092A). We would like to thank all participants who took part in the user study.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Declarations

Competing interests All the authors of this work declare that they have no conflicts of interest or competing interests regarding the publication of this paper.

References

1. Aggarwal, S., Goswami, D., Hooda, M., Chakravarty, A., Kar, A., et al. Recommendation systems for interactive multimedia entertainment. In: Data Visualization and Knowledge Engineering, pp.23–48. Springer, Cham, Switzerland (2020)
2. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: A review. Acm Computing Surveys (Csur) 43(3), 1–43 (2011)
3. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: A review. Acm Computing Surveys (Csur) 43(3), 1–43 (2011)
4. Alamdari, P.M., Navimipour, N.J., Hosseinzadeh, M., Safaei, A.A., Darwesh, A.: A systematic study on the recommender systems in the e-commerce. IEEE Access 8, 115694–115716 (2020)
5. Alem, L., Li, J.: A study of gestures in a video-mediated collaborative assembly task. Advances in Human-Computer Interaction **2011** (2011)
6. Alem, L., Li, J.: A study of gestures in a video-mediated collaborative assembly task. Advances in Human-Computer Interaction 2011 (2011)
7. Alibali, M.W., Nathan, M.J., Wolfgram, M.S., Church, R.B., Jacobs, S.A., Johnson Martinez, C., Knuth, E.J.: How teachers link ideas in mathematics instruction using speech and gesture: A corpus analysis. Cognition and instruction 32(1), 65–100 (2014)
8. Alibali, M.W., Nathan, M.J., Wolfgram, M.S., Church, R.B., Jacobs, S.A., Johnson Martinez, C., Knuth, E.J.: How teachers link ideas in mathematics instruction using speech and gesture: A corpus analysis. Cognition and instruction 32(1), 65–100 (2014)

9. Arthur, D., Vassilvitskii, S.: K-means++ the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp.1027–1035 (2007)
10. Bandura, A.: Observational learning. The international encyclopedia of communication (2008)
11. Bandura, A.: Social foundation of thought and action. Englewood cliffs, NJ: prentice Hall (1986)
12. Beddiar, D.R., Nini, B., Sabokrou, M., Hadid, A.: Vision-based human activity recognition: a survey. Multimedia Tools and Applications 79(41), 30509–30555 (2020)
13. Beddiar, D.R., Nini, B., Sabokrou, M., Hadid, A.: Vision-based human activity recognition: a survey. Multimedia Tools and Applications 79(41), 30509–30555 (2020)
14. Bétrancourt, M., Benetos, K.: Why and when does instructional video facilitate learning? a commentary to the special issue "developments and trends in learning with instructional video". Computers in Human Behavior 89, 471–475 (2018)
15. Bonaventura, X., Feixas, M., Sbert, M., Chuang, L., Wallraven, C.: A survey of viewpoint selection methods for polygonal models. Entropy 20(5), 370 (2018)
16. Boucheix, J.-M., Gauthier, P., Fontaine, J.-B., Jaffeux, S.: Mixed camera viewpoints improve learning medical hand procedure from video in nurse training? Computers in Human Behavior 89, 418–429 (2018)
17. Buckingham, G., Wong, J.D., Tang, M., Gribble, P.L., Goodale, M.A.: Observing object lifting errors modulates cortico-spinal excitability and improves object lifting performance. Cortex 50, 115–124 (2014)
18. Burris, A.: A child's-eye view: An examination of point-of-view camera use in four informal education settings. Visitor Studies 20(2), 218–237 (2017)
19. Burris, A.: A child's-eye view: An examination of point-of-view camera use in four informal education settings. Visitor Studies 20(2), 218–237 (2017)
20. Cernekova, Z., Pitas, I., Nikou, C.: Information theory-based shot cut/fade detection and video summarization. IEEE Transactions on circuits and systems for video technology 16(1), 82–91 (2005)
21. Clark, H.H., Brennan, S.E.: Grounding in communication. (1991)
22. Da'u, A., Salim, N.: Recommendation system based on deep learning methods: a systematic review and new directions. Artificial Intelligence Review 53(4), 2709–2748 (2020)
23. Da'u, A., Salim, N.: Recommendation system based on deep learning methods: a systematic review and new directions. Artificial Intelligence Review 53(4), 2709–2748 (2020)
24. de Koning, B.B., Marcus, N., Brucker, B., Ayres, P.: Does observing hand actions in animations and static graphics differentially affect learning of hand-manipulative tasks? Computers & Education 141, 103636 (2019)
25. de Koning, B.B., Hoogerheide, V., Boucheix, J.-M.: Developments and trends in learning with instructional video. Computers in Human Behavior (2018)
26. de Koning, B.B., Marcus, N., Brucker, B., Ayres, P.: Does observing hand actions in animations and static graphics differentially affect learning of hand-manipulative tasks? Computers & Education 141, 103636 (2019)
27. Deinzer, F., Denzler, J., Niemann, H.: Viewpoint selection–planning optimal sequences of views for object recognition. In: International Conference on Computer Analysis of Images and Patterns, pp.65–73 (2003). Springer
28. Endsley, M.R.: Measurement of situation awareness in dynamic systems. Human factors 37(1), 65–84 (1995)
29. Feixas, M., Sbert, M., González, F.: A unified information-theoretic framework for viewpoint selection and mesh saliency. ACM Transactions on Applied Perception (TAP) 6(1), 1–23 (2009)
30. Fiorella, L., Mayer, R.E.: What works and doesn't work with instructional video. Elsevier (2018)
31. Fiorella, L., van Gog, T., Hoogerheide, V., Mayer, R.E.: It's all a matter of perspective: Viewing first-person video modeling examples promotes learning of an assembly task. Journal of Educational Psychology 109(5), 653 (2017)
32. Fiorella, L., van Gog, T., Hoogerheide, V., Mayer, R.E.: It's all a matter of perspective: Viewing first-person video modeling examples promotes learning of an assembly task. Journal of Educational Psychology 109(5), 653 (2017)
33. Fiorella, L., Mayer, R.E.: What works and doesn't work with instructional video. Elsevier (2018)
34. Freitag, S., Weyers, B., Bönsch, A., Kuhlen, T.W.: Comparison and evaluation of viewpoint quality estimation algorithms for immersive virtual environments. ICAT-EGVE 15, 53–60 (2015)
35. Freitag, S., Weyers, B., Bönsch, A., Kuhlen, T.W.: Comparison and evaluation of viewpoint quality estimation algorithms for immersive virtual environments. ICAT-EGVE 15, 53–60 (2015)
36. Fussell, S.R., Setlock, L.D., Yang, J., Ou, J., Mauer, E., Kramer, A.D.: Gestures over video streams to support remote collaboration on physical tasks. Human-Computer Interaction 19(3), 273–309 (2004)
37. Graf, S., Herbig, T., Buck, M., Schmidt, G.: Features for voice activity detection: a comparative analysis. EURASIP Journal on Advances in Signal Processing 2015(1), 1–15 (2015)

38. Graf, S., Herbig, T., Buck, M., Schmidt, G.: Features for voice activity detection: a comparative analysis. EURASIP Journal on Advances in Signal Processing 2015(1), 1–15 (2015)
39. Guo, P.J., Kim, J., Rubin, R.: How video production affects student engagement: An empirical study of mooc videos. In: Proceedings of the First ACM Conference on Learning@ Scale Conference, pp.41–50 (2014)
40. Harris, D., Vine, S., Wilson, M., McGrath, J.S., LeBel, M., Buckingham, G.: Action observation for sensorimotor learning in surgery. Journal of British Surgery 105(13), 1713–1720 (2018)
41. Hart, S.G., Staveland, L.E.: Development of nasa-tlx (task load index): Results of empirical and theoretical research. In: Advances in Psychology vol. 52, pp.139–183. Elsevier, Los Angeles, CA (1988)
42. Henderson, M.L., Schroeder, N.L.: A systematic review of instructor presence in instructional videos: Effects on learning and affect. Computers and Education Open 2, 100059 (2021)
43. Henderson, M.L., Schroeder, N.L.: A systematic review of instructor presence in instructional videos: Effects on learning and affect. Computers and Education Open 2, 100059 (2021)
44. Heyes, C., Foster, C.: Motor learning by observation: evidence from a serial reaction time task. The Quarterly Journal of Experimental Psychology Section A 55(2), 593–607 (2002)
45. Heyes, C., Foster, C.: Motor learning by observation: evidence from a serial reaction time task. The Quarterly Journal of Experimental Psychology Section A 55(2), 593–607 (2002)
46. Hodges, N.J., Williams, A.M., Hayes, S.J., Breslin, G.: What is modelled during observational learning? Journal of sports sciences 25(5), 531–545 (2007)
47. Hodges, N.J., Williams, A.M., Hayes, S.J., Breslin, G.: What is modelled during observational learning? Journal of sports sciences 25(5), 531–545 (2007)
48. Huang, C., Gao, F., Pan, J., Yang, Z., Qiu, W., Chen, P., Yang, X., Shen, S., Cheng, K.-T.: Act: An autonomous drone cinematography system for action scenes. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp.7039–7046 (2018). IEEE
49. Huang, K., Li, J., Sousa, M., Grossman, T.: Immersivepov: Filming how-to videos with a head-mounted 360 action camera. In: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, pp.1–13 (2022)
50. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Transactions on Information Systems (TOIS) 20(4), 422–446 (2002)
51. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Transactions on Information Systems (TOIS) 20(4), 422–446 (2002)
52. Jiang, H., Wang, B., Wang, X., Christie, M., Chen, B.: Example-driven virtual cinematography by learning camera behaviors. ACM Transactions on Graphics (TOG) 39(4), 45–1 (2020)
53. Joo, H., Simon, T., Li, X., Liu, H., Tan, L., Gui, L., Banerjee, S., Godisart, T., Nabbe, B., Matthews, I., et al. Panoptic studio: A massively multiview system for social interaction capture. IEEE transactions on pattern analysis and machine intelligence 41(1), 190–204 (2017)
54. Kamada, T., Kawai, S.: A simple method for computing general position in displaying three-dimensional objects. Computer Vision, Graphics, and Image Processing 41(1), 43–56 (1988)
55. Kamada, T., Kawai, S.: A simple method for computing general position in displaying three-dimensional objects. Computer Vision, Graphics, and Image Processing 41(1), 43–56 (1988)
56. Key, M.R.: The relationship of verbal and nonverbal communication. In: The Relationship of Verbal and Nonverbal Communication. De Gruyter Mouton, Berlin (2011)
57. Koenderink, J.J., van Doorn, A.J.: The internal representation of solid shape with respect to vision. Biological cybernetics 32(4), 211–216 (1979)
58. Koenderink, J.J., van Doorn, A.J.: The internal representation of solid shape with respect to vision. Biological cybernetics 32(4), 211–216 (1979)
59. Kumar, P., Thakur, R.S.: Recommendation system techniques and related issues: a survey. International Journal of Information Technology 10(4), 495–501 (2018)
60. Kumar, P., Thakur, R.S.: Recommendation system techniques and related issues: a survey. International Journal of Information Technology 10(4), 495–501 (2018)
61. Laporte, C., Arbel, T.: Efficient discriminant viewpoint selection for active bayesian recognition. International Journal of Computer Vision 68(3), 267–287 (2006)
62. Laporte, C., Arbel, T.: Efficient discriminant viewpoint selection for active bayesian recognition. International Journal of Computer Vision 68(3), 267–287 (2006)
63. Larkin, K.G.: Reflections on shannon information: In search of a natural information-entropy for images. arXiv preprint arXiv:1609.01117 (2016)
64. Leu, M.C., ElMaraghy, H.A., Nee, A.Y., Ong, S.K., Lanzetta, M., Putz, M., Zhu, W., Bernard, A.: Cad model based virtual assembly simulation, planning and training. CIRP Annals 62(2), 799–822 (2013)
65. Leu, M.C., ElMaraghy, H.A., Nee, A.Y., Ong, S.K., Lanzetta, M., Putz, M., Zhu, W., Bernard, A.: Cad model based virtual assembly simulation, planning and training. CIRP Annals 62(2), 799–822 (2013)

66. Lino, C., Christie, M., Ranon, R., Bares, W.: The director's lens: an intelligent assistant for virtual cinematography. In: Proceedings of the 19th ACM International Conference on Multimedia, pp.323–332 (2011)
67. Lipowski, Z.J.: Sensory and information inputs overload: behavioral effects. Comprehensive Psychiatry (1975)
68. Mason, S., et al. Heuristic reasoning strategy for automated sensor placement. Photogrammetric engineering and remote sensing 63(9), 1093–1101 (1997)
69. Mavlankar, A., Agrawal, P., Pang, D., Halawa, S., Cheung, N.-M., Girod, B.: An interactive region-of-interest video streaming system for online lecture viewing. In: 2010 18th International Packet Video Workshop, pp.64–71 (2010). IEEE
70. Mayer, R.E.: Evidence-based principles for how to design effective instructional videos. Journal of Applied Research in Memory and Cognition (2021)
71. Mayer, R.E.: Introduction to multimedia learning. (2014)
72. Mayer, R.E.: Principles based on social cues in multimedia learning: Personalization, voice, image, and embodiment principles. The Cambridge handbook of multimedia learning 16, 345–370 (2014)
73. Mayer, R.E.: Evidence-based principles for how to design effective instructional videos. Journal of Applied Research in Memory and Cognition (2021)
74. Mayer, R.E., Fiorella, L., Stull, A.: Five ways to increase the effectiveness of instructional video. Educational Technology Research and Development 68(3), 837–852 (2020)
75. McNeill, D.: Hand and Mind. De Gruyter Mouton, Berlin (2011)
76. Mittelberg, I., Evola, V.: 131. iconic and representational gestures. In: Handbücher zur Sprach-und Kommunikationswissenschaft/Handbooks of Linguistics and Communication Science (HSK) 38/2, pp.1732–1746. De Gruyter Mouton, Berlin (2014)
77. Montes, J., Figueroa, P.: Vr salsa: learning to dance in virtual reality. In: Proceedings of the IX Latin American Conference on Human Computer Interaction, pp.1–4 (2019)
78. Munea, T.L., Jembre, Y.Z., Weldegebriel, H.T., Chen, L., Huang, C., Yang, C.: The progress of human pose estimation: a survey and taxonomy of models applied in 2d human pose estimation. IEEE Access 8, 133330–133348 (2020)
79. Nocedal, J., Wright, S.J.: Quasi-newton methods. Numerical optimization, 135–163 (2006)
80. Nocedal, J., Wright, S.J.: Quasi-newton methods. Numerical optimization, 135–163 (2006)
81. Novack, M.A., Goldin-Meadow, S.: Gesture as representational action: A paper about function. Psychonomic Bulletin & Review 24(3), 652–665 (2017)
82. Olagoke, A.S., Ibrahim, H., Teoh, S.S.: Literature survey on multi-camera system and its application. IEEE Access 8, 172892–172922 (2020)
83. Oliva, A., Torralba, A.: The role of context in object recognition. Trends in cognitive sciences 11(12), 520–527 (2007)
84. Ou, C., Joyner, D.A., Goel, A.K.: Designing and developing video lessons for online learning: A seven-principle model. Online Learning 23(2), 82–104 (2019)
85. Pareek, P., Thakkar, A.: A survey on video-based human action recognition: recent updates, datasets, challenges, and applications. Artificial Intelligence Review 54(3), 2259–2322 (2021)
86. Pareek, P., Thakkar, A.: A survey on video-based human action recognition: recent updates, datasets, challenges, and applications. Artificial Intelligence Review 54(3), 2259–2322 (2021)
87. Pi, Z., Hong, J., Yang, J.: Does instructor's image size in video lectures affect learning outcomes? Journal of Computer Assisted Learning 33(4), 347–354 (2017)
88. Pi, Z., Xu, K., Liu, C., Yang, J.: Instructor presence in video lectures: Eye gaze matters, but not body orientation. Computers & Education 144, 103713 (2020)
89. Rahimian, C., Kearney, J.K.: Optimal camera placement for motion capture systems. IEEE transactions on visualization and computer graphics 23(3), 1209–1221 (2016)
90. Rahimian, C., Kearney, J.K.: Optimal camera placement for motion capture systems. IEEE transactions on visualization and computer graphics 23(3), 1209–1221 (2016)
91. Rahnert, K.: The teaching hand in remote accounting education: bringing mirror neurons into the debate. Accounting Education 31(5), 482–501 (2022)
92. Razlighi, Q., Kehtarnavaz, N.: A comparison study of image spatial entropy. In: Visual Communications and Image Processing 2009, vol. 7257, p.72571 (2009). International Society for Optics and Photonics
93. Rehatschek, H.: Experiences from the introduction of an automated lecture recording system. In: International Conference on Interactive Collaborative Learning, pp.151–162 (2018). Springer
94. Ristad, E.S., Yianilos, P.N.: Learning string-edit distance. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(5), 522–532 (1998)
95. Ristad, E.S., Yianilos, P.N.: Learning string-edit distance. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(5), 522–532 (1998)

96. Rosen, M.A., Salas, E., Pavlas, D., Jensen, R., Fu, D., Lampton, D.: based training: A review of instructional features. Human factors 52(5), 596–609 (2010)
97. Rosen, M.A., Salas, E., Pavlas, D., Jensen, R., Fu, D., Lampton, D.: based training: A review of instructional features. Human factors 52(5), 596–609 (2010)
98. Rust, N.C., Stocker, A.A.: Ambiguity and invariance: two fundamental challenges for visual processing. Current opinion in neurobiology 20(3), 382–388 (2010)
99. Rust, N.C., Stocker, A.A.: Ambiguity and invariance: two fundamental challenges for visual processing. Current opinion in neurobiology 20(3), 382–388 (2010)
100. Sablić, M., Mirosavljević, A., Škugor, A.: Video-based learning (vbl)–past, present and future: An overview of the research published from 2008 to 2019. Technology, Knowledge and Learning 26(4), 1061–1077 (2021)
101. Sablić, M., Mirosavljević, A., Škugor, A.: Video-based learning (vbl)–past, present and future: An overview of the research published from 2008 to 2019. Technology, Knowledge and Learning 26(4), 1061–1077 (2021)
102. Sakane, S., Niepold, R., Sato, T., Shirai, Y.: Illumination setup planning for a hand-eye system based on an environmental model. Advanced Robotics 6(4), 461–482 (1991)
103. Sakane, S., Ish, M., Kakikura, M.: Occlusion avoidance of visual sensors based on a hand-eye action simulator system: Heaven. Advanced robotics 2(2), 149–165 (1987)
104. Sakane, S., Niepold, R., Sato, T., Shirai, Y.: Illumination setup planning for a hand-eye system based on an environmental model. Advanced Robotics 6(4), 461–482 (1991)
105. Sasikumar, P., Chittajallu, S., Raj, N., Bai, H., Billinghurst, M.: Spatial perception enhancement in assembly training using augmented volumetric playback. Frontiers in Virtual Reality, 100 (2021)
106. Sasikumar, P., Chittajallu, S., Raj, N., Bai, H., Billinghurst, M.: Spatial perception enhancement in assembly training using augmented volumetric playback. Frontiers in Virtual Reality, 100 (2021)
107. Schurgin, M.W., Flombaum, J.I.: Exploiting core knowledge for visual object recognition. Journal of Experimental Psychology: General 146(3), 362 (2017)
108. Secord, A., Lu, J., Finkelstein, A., Singh, M., Nealen, A.: Perceptual models of viewpoint preference. ACM Transactions on Graphics (TOG) 30(5), 1–12 (2011)
109. Shannon, C.E.: A mathematical theory of communication. The Bell system technical journal 27(3), 379–423 (1948)
110. Takeuchi, Y., Ohnishi, N., Sugie, N.: Active vision system based on information theory. Systems and Computers in Japan 29(11), 31–39 (1998)
111. Takeuchi, Y., Ohnishi, N., Sugie, N.: Active vision system based on information theory. Systems and Computers in Japan 29(11), 31–39 (1998)
112. Tang, R., Yang, X.-D., Bateman, S., Jorge, J., Tang, A.: Physio@ home: Exploring visual guidance and feedback techniques for physiotherapy exercises. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp.4123–4132 (2015)
113. Tarabanis, K.A., Allen, P.K., Tsai, R.Y.: A survey of sensor planning in computer vision. IEEE transactions on Robotics and Automation 11(1), 86–104 (1995)
114. Tarabanis, K.A., Tsai, R.Y., Allen, P.K.: The mvp sensor planning system for robotic vision tasks. IEEE Transactions on Robotics and Automation 11(1), 72–85 (1995)
115. Tarabanis, K.A., Allen, P.K., Tsai, R.Y.: A survey of sensor planning in computer vision. IEEE transactions on Robotics and Automation 11(1), 86–104 (1995)
116. Van der Meij, H.: Reviews in instructional video. Computers & education 114, 164–174 (2017)
117. van Wermeskerken, M., Ravensbergen, S., van Gog, T.: Effects of instructor presence in video modeling examples on attention and learning. Computers in Human Behavior 89, 430–438 (2018)
118. van Wermeskerken, M., van Gog, T.: Seeing the instructor's face and gaze in demonstration video examples affects attention allocation but not learning. Computers & Education 113, 98–107 (2017)
119. van Wermeskerken, M., van Gog, T.: Seeing the instructor's face and gaze in demonstration video examples affects attention allocation but not learning. Computers & Education 113, 98–107 (2017)
120. van Wermeskerken, M., Ravensbergen, S., van Gog, T.: Effects of instructor presence in video modeling examples on attention and learning. Computers in Human Behavior 89, 430–438 (2018)
121. Vázquez, P.-P., Feixas, M., Sbert, M., Heidrich, W.: Viewpoint selection using viewpoint entropy. In: VMV, vol. 1, pp.273–280 (2001). Citeseer
122. Vázquez, P.-P., Feixas, M., Sbert, M., Llobet, A.: Realtime automatic selection of good molecular views. Computers & Graphics 30(1), 98–110 (2006)
123. Vázquez, P.-P., Feixas, M., Sbert, M., Llobet, A.: Viewpoint entropy: a new tool for obtaining good views of molecules. In: ACM International Conference Proceeding Series, vol. 22, pp.183–188 (2002)
124. Vázquez, P.-P., Feixast, M., Sbert, M., Heidrich, W.: Image-based modeling using viewpoint entropy. In: Advances in Modelling, Animation and Rendering, pp.267–279. Springer, London (2002)

125. Vázquez, P.-P., Sbert, M.: Automatic indoor scene exploration. In: Proceedings of 6th International Conference on Computer Graphics and Artificial Intelligence 3IA, pp.13–24 (2003). Citeseer
126. Vázquez, P.-P., Feixast, M., Sbert, M., Heidrich, W.: Image-based modeling using viewpoint entropy. In: Advances in Modelling, Animation and Rendering, pp. 267–279. Springer, London (2002)
127. Vázquez, P.-P., Feixas, M., Sbert, M., Llobet, A.: Realtime automatic selection of good molecular views. Computers & Graphics 30(1), 98–110 (2006)
128. Wang, Z., Bai, X., Zhang, S., Billinghurst, M., He, W., Wang, P., Lan, W., Min, H., Chen, Y.: A comprehensive review of augmented reality-based instruction in manual assembly, training and repair. Robotics and Computer-Integrated Manufacturing 78, 102407 (2022)
129. Wang, Y., Wang, L., Li, Y., He, D., Chen, W., Liu, T.-Y.: A theoretical analysis of ndcg ranking measures. In: Proceedings of the 26th Annual Conference on Learning Theory (COLT 2013), vol. 8, p.6 (2013). Citeseer
130. Wang, Z., Bai, X., Zhang, S., Billinghurst, M., He, W., Wang, P., Lan, W., Min, H., Chen, Y.: A comprehensive review of augmented reality-based instruction in manual assembly, training and repair. Robotics and Computer-Integrated Manufacturing 78, 102407 (2022)
131. Yang, X., Guo, Y., Liu, Y., Steck, H.: A survey of collaborative filtering based social recommender systems. Computer communications 41, 1–10 (2014)
132. Yoo, J.E., Seo, K., Park, S., Kim, J., Lee, D., Noh, J.: Virtual camera layout generation using a reference video. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp.1–11 (2021)
133. Zhang, C., Rui, Y., Crawford, J., He, L.-W.: An automated end-to-end lecture capture and broadcasting system. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 4(1), 1–23 (2008)
134. Zhang, X.-L., Wu, J.: Deep belief networks based voice activity detection. IEEE Transactions on Audio, Speech, and Language Processing 21(4), 697–710 (2012)
135. Zhang, X.-L., Wu, J.: Deep belief networks based voice activity detection. IEEE Transactions on Audio, Speech, and Language Processing 21(4), 697–710 (2012)

Authors and Affiliations

Saugata Biswas[1] · Ernst Kruijff[1] · Eduardo Veas[2]

Ernst Kruijff
ernst.kruijff@h-brs.de

Eduardo Veas
eveas@tugraz.at

1 Department of Computer Science, Bonn-Rhein-Sieg University of Applied Sciences, Grantham-Allee 20, Sankt Augustin 53757, Germany

2 Institute of Interactive Systems and Data Science, Graz University of Technology, Sandgasse 36/III, Graz 8010, Austria