SceneCam: Improving Multi-Camera Remote Collaboration using Augmented Reality

Troels A. Rasmussen *
Department of Computer Science, Aarhus University, Denmark

Weidong Huang†
University of Technology Sydney, Australia

ABSTRACT

Systems for remote collaboration on physical tasks generally use AR/VR technologies to create a shared visual space for collaborators to perform tasks together. The shared space often comes from a single camera view. Prior research has not reported on the benefits of using multiple cameras for remote collaboration. On the contrary, there seems to be some usability issues, which must be addressed, when designing remote collaboration systems that use multiple cameras to capture different areas and perspectives of a task space. To be usable, a multi-camera remote collaboration system must indicate to the local user which camera the remote user is looking at and vice versa, the system must make it fast and easy for the remote user to obtain the right camera view for a given collaborative task. We present SceneCam, an AR prototype with which we explore different techniques for improving the usability of multi-camera remote collaboration by making camera selection easier and faster. Specifically, SceneCam implements two camera selection techniques. The first technique nudges the remote user to manually select an optimal camera view of the local user's actions. The second technique automatically selects an optimal camera view of the local user and shows it to the remote user. Additionally, SceneCam implements two focus-in-context views (exocentric and egocentric views) that provide the remote user with a spatial overview of the local user's whereabouts in relation to the multiple task space areas and direct visual access to the camera views of said areas. Camera selection techniques (manual point-and-click, nudging, automatic), and focus-in-context views (no focus-in-context view, exocentric, egocentric) make up the two dimensions in a design space for multi-camera remote collaboration. We describe how SceneCam spans this design space. Lastly, as part of future work we discuss some hypotheses regarding the effects of the proposed camera selection techniques, focus-in-context views and combinations hereof on the usability of multi-camera remote collaboration.

Index Terms: Remote collaboration—Augmented reality—Multiple cameras—Usability;

1 INTRODUCTION: MULTI-CAMERA REMOTE COLLABORATION ON PHYSICAL TASKS

In some remote collaboration scenarios the physical task space can be separated into areas. For instance, our own observations of how service is carried out on machines in the manufacturing industry, specifically service on CNC machines and large inline printing machines, indicate that there are often multiple areas on a machine that the service technician must troubleshoot in collaboration with a remote helper, and sometimes the work carried out in one area is interdependent on work done in another area. A different scenario with similar characteristics is that of a researcher, who divides up his work space into smaller areas on the whiteboard and on his desk, while he collaborates with a remote researcher on a design problem or mathematical problem etc.

One approach to sharing multiple task space areas with a remote collaborator is to use multiple cameras - one or more cameras per task space area - which we will call *multi-camera remote collaboration on physical tasks.* Importantly, we want to make it clear to the reader that from hereon we refer to ordinary RGB cameras streaming video of the task space from the local user/worker to the remote user/helper. In this paper we are not dealing with RGB-D cameras, such as the Kinect, which has the ability to reconstruct the task space in 3D. It is important to make this distinction, because the interaction and usability challenges differ between the two technologies.

The research on multi-camera remote collaboration is sparse and discouraging, as multiple scene cameras seem to cause usability issues for the collaborators. Manual selection of the correct camera view is time consuming for the remote user, the remote user struggles to understand and remember the spatial relationship between task space areas, the scope of the shared visual resources is not clear, and finally the local user does not know which camera the remote user is looking at [3, 5].

We present the AR prototype SceneCam, a multi-camera remote collaboration system, which addresses some of the above usability issues using of a variety of camera selection techniques and focus-in-context views. To our knowledge, very few researchers have explored camera selection techniques and focus-in-context views for improving multi-camera remote collaboration and few have used AR technology as the means for the exploration.

SceneCam implements a context-aware algorithm that collects contextual information about the local user (location, orientation, gaze etc.) in relation to the task space areas and uses this information to predict the optimal camera view of the local user's actions in a task space. SceneCam implements two camera selection techniques, which rely on the context-aware algorithm: 1) nudging of camera selection, where the system uses the result of the context-aware algorithm to nudge the remote user to select the optimal camera view from a list, 2) automatic camera selection, where the system uses the result of the context-aware algorithm to automatically select the optimal camera view to show the remote user. In a nutshell, we hypothesize that these camera selection techniques will make the task of selecting an optimal camera view of the local user's actions less mentally demanding and faster than manually selecting between cameras from a list.

Poupyrev et al. [11] described virtual reality manipulation techniques using metaphors divided into two categories, exocentric and egocentric, which are two fundamentally different views for user interaction with a virtual environment. With the exocentric view, the user interacts with the virtual environment from the outside, as if he is looking down on it from a bird's eye view, whereas with an egocentric view the user interacts from a point of view inside the environment. Inspired by these metaphors, SceneCam implements exocentric and egocentric focus-in-context views in the remote user's interface. The purpose of a focus-in-context view is to show the spatial relationship between task space areas and the local user's movements in relation to them (context), while giving access to detailed views of the areas (focus). We hypothesize that a focus-

*e-mail: troels.rasmussen@cs.au.dk
†e-mail: weidong.huang@uts.edu.au

in-context view will make camera selection easier and faster than manually selecting between cameras from a list. For instance, if the local user refers to an area relative to his own point of view (”look at the area to the left of me”), the remote user can use the focus-in-context view to recognize the area and select a view of the area rather than have to recall the spatial relationship of the task space areas from memory.

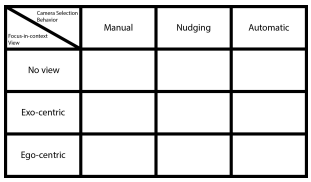

Figure 1: Design space for multi-camera remote collaboration

The camera selection techniques (manual point-and-click, nudging, automatic), and focus-in-context views (no focus-in-context view, exocentric, egocentric) make up the two dimensions in a design space for multi-camera remote collaboration (see figure 1). SceneCam spans the entire design space. We describe how SceneCam implements the techniques in more detail in section 3.

Finally, we discuss the limitations of SceneCam and future plans of conducting controlled experiments with SceneCam to understand the effect of the proposed camera selection techniques and focus-in-context views on multi-camera remote collaboration.

2 Related Work

Different approaches to sharing a task space with a remote user has been proposed, from using a handheld or head-mounted camera, thus capturing the task space from the point of view of the local user [3], to freezing the video from the handheld/head-mounted camera making it easier to annotate [1, 2], to using scene cameras, i.e. sharing views of the task space from one or more cameras mounted in the environment [5, 8], to sharing a 3D reconstruction of the task space [14], sometimes viewed by the remote user in VR [4, 15]. Given that our focus is multi-camera remote collaboration on physical tasks, we dedicate the remainder of this section to related work on this topic.

2.1 Shared visual space from multiple camera views

Researchers have investigated the usefulness of providing the remote user with views of the task space from multiple cameras [3, 5, 7].

For instance Gaver et al. [5] build an early task oriented media space, Multiple Target Video (MTV), where multiple scene cameras were pointed at objects and the task environment. The aim of MTV was to support focused collaboration on physical tasks across two office spaces. Gaver et al. found that their pairs of participants experienced difficulties using the multiple scene cameras, including difficulties establishing a shared frame of reference, understanding which parts of a space was visually accessible to the remote user, and switching between camera views. Similarly, Fussell et al. [3] found that simultaneously providing helpers with a view of a task space from a head-mounted camera (close-up view) and a scene mounted camera (over-the-shoulder view) did not improve collaboration performance of worker-helper pairs in comparison to just using the scene camera, possibly because the helpers spend too much time deciding what view to pay attention to.

So, using multiple cameras might seem like an unsuccessful path to pursue. However, in none of the above examples did the researchers make use of a focus-in-context view to simultaneously visualize the spatial relationship between and give access to the camera views. Nor did they make use of automatic selection of the optimal camera view. Rather, the helper ”jumped” from a view of one task space area to another by manually selecting a camera from a list of cameras, thus experiencing spatial discontinuities. Work addressing this issue include [9, 12, 16]. Ranjan et al. [12] compared a static scene camera that provided a wide-shot context view of the entire task space to a scene camera that automatically zoomed in on the task space area, where the worker's hands were, and at the same time provided a contextual overview showing the relationship between task space areas, whenever the worker moved his hands from one area to another. They found substantial performance benefits for the automatic system. Similarly, Norris et al. [9] implemented a focus-in-context video system, where zoom-functionality enabled the remote user to view multiple zoomed-in high-resolution areas in a low-resolution wide-angle view of the local user's entire task space. They found that the spatially connected detailed views embedded in the contextual overview helped with view reconciliation, i.e. made it easier to collaborate using multiple views.

With the SceneCam prototype we also wish to embed detailed views of task space areas in a contextual overview of the task space, and we enable automatic selection of the optimal view based on the worker's whereabouts. However, by using multiple cameras we are not limited to obtaining detailed views from one viewpoint as in previous work.

2.2 Egocentric and exocentric views in AR/VR

Examples of a user with an exocentric view of an environment collaborating with another user in an egocentric MR view of the same environment can be found in [6, 13]. In [13] researchers used a 2D map on a tabletop as an exocentric view of an outdoor environment. They demonstrated a technique where an indoor user could place his hands and other three-dimensional props on the 2D map. Hands and objects were then 3D reconstructed and visualized to an outdoor AR-user at the corresponding location in the real environment. This technique was used for navigation and layout planning.

SceneCam is to our knowledge the first prototype that demonstrates the use of exocentric and egocentric views to navigate between a contextual overview of a task space and detailed views of task space areas captured by multiple cameras.

3 SceneCam: Multi-camera AR Remote Collaboration

3.1 The core functionality of SceneCam

We present SceneCam, a multi-camera AR remote collaboration prototype. Scene cameras - i.e. tablet cameras, smartphone cameras or webcams mounted in the environment of the local user's task space - provide a remote user with multiple views of the task space in the form of live video feeds. Using a 2D screen interface (PC / tablet), the remote user can select between video feeds and point and sketch on the currently selected video feed. The 2D pointing gestures and sketches are interpreted in 3D, placed directly in the task space and shown to the local user on an AR-HMD (in our case a Microsoft Hololens with inside-out tracking). Thus, SceneCam gives AR capabilities to ordinary RBG scene cameras. This works by getting the local user to track the pose of an AR marker on a scene camera thereby aligning a virtual model of the scene camera to the real camera. See figure 2 showing four scene cameras with virtual models aligned to them. From knowing the intrinsics of the scene cameras and pose of the corresponding virtual cameras in the world coordinate system of the Hololens, it is possible for the AR application to interpret 2D annotations made on the video feeds in 3D using using the spraypaint technique [10]. Audio communication is bidirectional. See figure 3 and 4 for screenshots of the remote user's 2D screen interface and local user's AR interface. See figure 5 for an illustration of where the core functionality of SceneCam is placed in the design space.

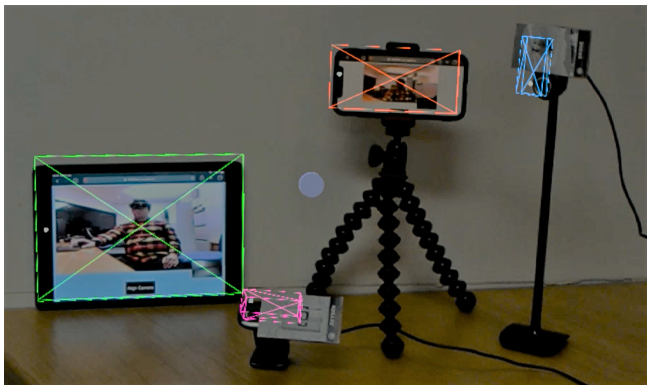

Figure 2: Screenshot from point of view of AR-HMD. Four scene cameras of different kinds with virtual camera models aligned to them. From left to right: iPad, Logitech 270p webcam, iPhone 10, Logitech webcam.

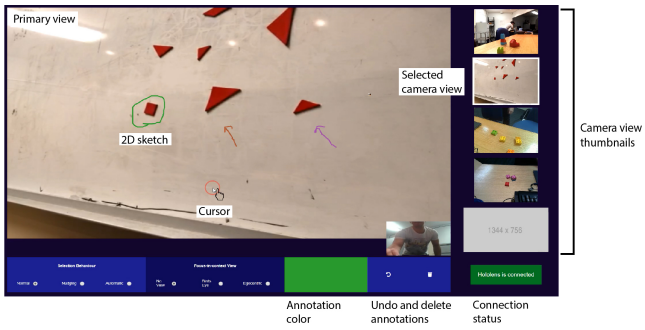

Figure 3: Core elements of remote user's 2D screen interface. Live video from each of the scene cameras capturing the local user's task space are made available as thumbnails in a vertical list in the remote user's 2D screen interface (on desktop/tablet). SceneCam currently supports up to five scene cameras at a time, and in this case four scene cameras are connected. The remote user can click on a video in the list of thumbnails to select it as the primary view. The remote user can make 2D annotations (pointing gestures and sketches) on the primary view. In this case the remote user draws two arrows and a circle to point to some puzzle pieces on a whiteboard. The AR application running on the local user's AR-HMD interprets the remote user's 2D annotations as 3D annotations and places them on the whiteboard using the spraypaint technique (see figure 4 for the result).

SceneCam spans the design space for multi-camera remote collaboration in figure 1 and thus contains example implementations of nudging of manual camera selection, automatic camera selection and exocentric/egocentric focus-in-context views for camera selection. Below we describe the example implementations in more detail.

3.2 Context-aware camera selection algorithm

An algorithm running on the local user's AR-HMD collects contextual information about the pose of the local user's head in relation to the pose of the task space areas and scene cameras. The algorithm uses this information to make inferences about a local user's engagement in a task space area and to decide which one of the scene cameras, if any, captures the optimal view of his actions. The resulting optimal view is then used by either the automatic camera selection technique, which as the name implies selects the optimal view to be the primary view, or by the nudging camera selection technique, which visually emphasizes the optimal view in the list of views to nudge the remote user to select it as the primary view.

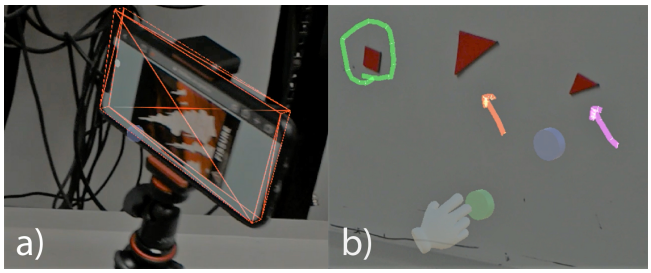

Figure 4: SceneCam interface from the point of view of the local AR user. a) A 3D model of a virtual camera is aligned to a real camera, in this case a smartphone, by tracking a marker on the phone once. We use the standard Vuforia markers. b) Annotations, drawn by a remote user, are interpreted in 3D and shown to the local user in AR directly on the whiteboard.

Camera Selection Behavior / Focus-in-context View	Manual	Nudging	Automatic
No view	●		
Exo-centric			
Ego-centric			

Figure 5: Position of core functionality in the design space.

Here is how the algorithm works. A task space area is described geometrically as a 3D point, where a virtual scene camera ray intersects the mesh reconstruction of the task space. Thus, there is one task space area associated with each virtual scene camera - a (scene camera, task space area)-pair. See pseudo-algorithm 1 for detailed steps on finding (scene camera, task space area)-pairs. These steps are executed whenever a virtual scene camera is aligned to a real scene camera.

foreach *scene camera* **do**
  create a ray with origin in the focal point of the scene camera and direction along the z-axis of the scene camera;
  if *ray intersects the mesh reconstruction of the task space* **then**
    create task space area defined as (position at intersection point, surface normal at intersection point);
    visualize task space area in AR;
    add (scene camera, task space area)-pair to list of pairs;
end
return list of (scene camera, task space area)-pairs;

Algorithm 1: Finding (scene camera, task space area)-pairs.

After collecting a list of (scene camera, task space area)-pairs, the algorithm runs through the list to find the optimal view of the local user's actions. Either one of the scene cameras is currently capturing the optimal view of the local user's actions or the focus-in-context view is (if any). See pseudo-algorithm 2 for detailed steps on finding the optimal view. These steps are executed continuously as the local user moves around the task space and the optimal view changes.

The occlusion score of a scene camera is simply calculated as the proportion of the camera's view taken up by the local user's head: (area of head in screen space / area of camera video).

In figure 6 we have illustrated three different situations the local user can find himself in which influence the algorithm of the context-aware camera selection.

foreach *(scene camera, task space area)-pair* **do**
  calculate distance between focal point of scene camera and position of task space area;
  create semi-sphere with center in the position of the task space area, facing in the direction of the surface normal of the task space area and with radius proportional to the distance between focal point of scene camera and position of task space area;
  if *local user is inside the semi-sphere* **then**
    add (scene camera, task space area) to list of candidates for optimal view;
end
if *list of candidates has size 0* **then**
  return focus-in-context view;
else if *list of candidates has size 1* **then**
  return the scene camera of the candidate;
else if *list of candidates has size larger than 1* **then**
  foreach *candidate* **do**
    calculate occlusion score;
  end
  return the scene camera of the candidate with the lowest occlusion score;

Algorithm 2: Finding the optimal view.

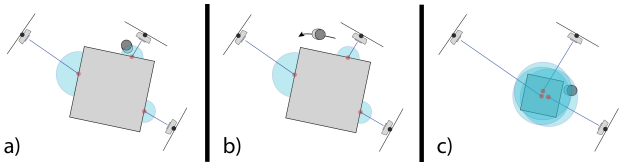

Figure 6: Three situations of the local user influencing the algorithm of context-aware camera selection (seen from above). a) The local user is inside the semi-sphere of a task space area. The scene camera associated with the task space area captures the optimal view of the local user's actions. b) The local user is not inside the semi-sphere of any task space areas, because he is transitioning from one area to another. The optimal view is the focus-in-context view, if any. c) The local user is inside multiple semi-spheres, because multiple scene cameras capture the same task space area from different perspectives. The optimal view must be decided using an occlusion score.

3.3 Nudging and automatic camera selection

The context-aware camera selection algorithm passes its estimate of the optimal view of the local user to either the nudging or automatic camera selection techniques. The nudging technique implemented in SceneCam is to simply highlight the optimal view in the list of camera view thumbnails using a red colored border. While this implementation leaves room for aesthetic improvement, it should make it clear to the remote user, which camera view is recommended by SceneCam. See figure 8 showing a screenshot of the nudging technique as implemented in the SceneCam prototype.

Automatic camera selection mimics the behaviour of a remote user selecting a camera view from the list of thumbnails. Hence, the camera view selected as the optimal view by the context-aware camera selection algorithm is automatically made the primary view, and its thumbnail is highlighted using a white border. See figure 7 showing the position of the nudging and automatic camera selection techniques in the design space.

3.4 Exo- and egocentric focus-in-context views

A context view of a task space provides the remote user with a spatial overview of the task space areas, cameras and the whereabouts of the local user in relation to them. A focus view is a close-up camera view of a task space area and the local user's object manipulations in

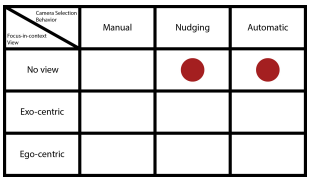

Figure 7: Position of nudging and automatic camera selection technique in the design space.

Figure 8: Nudging camera selection technique. Zoomed-in view of camera view thumbnails in remote user's interface, where the optimal view - according to the context-aware camera selection algorithm - is highlighted using a red border, while the currently selected camera view is highlighted using a white border.

the area. When a task space consists of multiple spatially distributed areas, the idea of a multi-camera setup is to capture at least one focus view of each area, and possibly two or more focus views from different perspectives. A focus-in-context view is an interactive context view with focus views mapped to the locations of the task space areas in the context view. It is used by the remote user to manually select and navigate between focus views. The aim of a focus-in-context view is to make it easier for the remote user to understand the spatial relationship between task space areas and the local user, while enabling quick and easy navigation between focus views. A focus-in-context view addresses the usability issues that arise from presenting camera views side by side in a list. A list of camera views, in comparison to a focus-in-context view, contains no information about the spatial relationship between the task space areas and the local user and thus makes the decision process of selecting an appropriate camera view more mentally demanding of the remote user.

SceneCam contains an implementation of both an egocentric and exocentric focus-in-context view in the remote user's interface. See figure 9 for an overview of the multi-camera setup used to demonstrate the implementation of the focus-in-context views. The exocentric view is a virtual bird's eye view of the position and orientation of the scene cameras, task space areas and the local user. See figure 10 for a screenshot of the exocentric view in the remote user's interface. The (scene camera, task space area)-pairs, represented by rectangular and circular icons respectively, are dynamically added to the exocentric view, whenever a virtual scene camera is aligned to a real scene camera by scanning a marker on the camera. The local user's position and orientation, represented by a triangular icon, is live updated in the exocentric view. Hovering over an icon of a task space area with the mouse reveals a thumbnail of the view from the associated scene camera, and clicking on the task space area selects the view from the scene camera as the primary view.

For the implementation of the egocentric view a wide-shot scene camera is assigned by the local user to capture an overview of the

Figure 9: Overview of multi-camera setup used to demonstrate the implementation of exo- and egocentric focus-in-context views. Camera 1 is a phone camera pointed at the whiteboard on which some magnetic puzzle pieces have been placed. Camera 2 is a webcam pointing to an area on the desk with some LEGO bricks. Camera 3 is a webcam pointing to another area on the desk with some LEGO bricks.

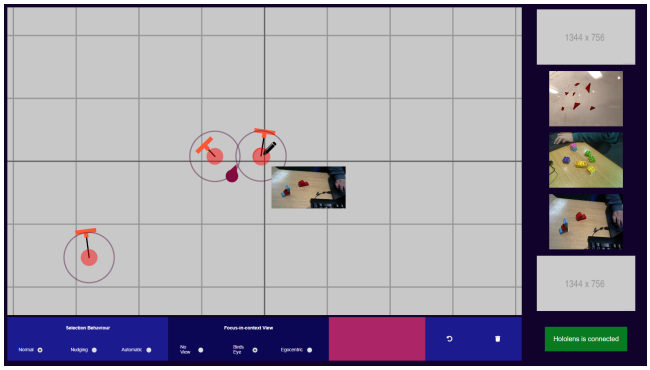

Figure 10: Screenshot of exocentric focus-in-context view. The position of the local user (the pink head icon) is updated live in relation of the task space areas (circles with red center) and scene cameras (orange squares). Upon hovering over the center of a task space area, a thumbnail of the camera view of the area is shown. Upon clicking on a task space area, the camera view of the area is made the primary view.

task space, and thus must be placed in such a way that it captures the task space areas and the local user. The egocentric view is augmented with interactive pinpoint needles on the task space areas. Upon hovering over a pinpoint needle on a task space area, a thumbnail shows the view from the associated scene camera. By clicking on a pinpoint needle, the remote user makes the view of the associated scene camera the primary view, and he is now able to point and sketch on the view. See figure 11 for a screenshot of the egocentric view in the remote user's interface, and figure 12 for the position of the exo- and egocentric views in the design space.

3.5 Combining camera selection techniques and focus-in-context views

As is evident from the design space (see figure 13), SceneCam combines the behaviour of nudging and automatic camera selection with the focus-in-context views. When nudging is combined with either an ego- or exocentric focus-in-context view, SceneCam will nudge the selection of the focus-in-context view, whenever the local user is not inside any of the task space areas (i.e. not inside any semi-spheres). When automatic camera selection is combined with either an ego- or exocentric focus-in-context view, the focus-in-context view will automatically appear, whenever the local user is not inside any of the task space areas. This view selection behaviour is similar to how the pan-zoom-tilt camera would zoom out when the local user's hand transitioned from one area to another in [12].

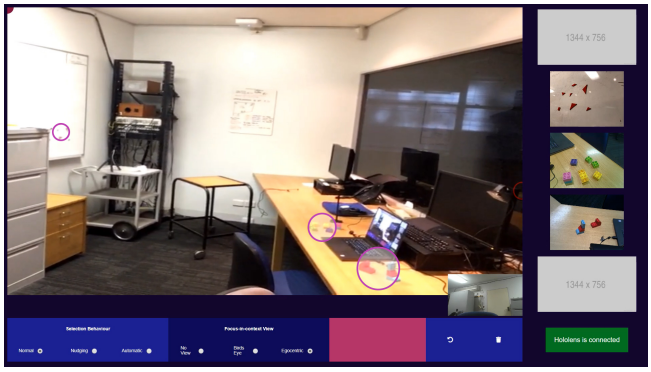

Figure 11: Screenshot of egocentric focus-in-context view. A video feed (in this case from an iPad camera) provides an overview of the task space augmented with "pinpoint needles" at the positions of the task space areas. Upon hovering over a pinpoint needle, a thumbnail of the camera view of the area is shown. Upon clicking on a pinpoint needle, the camera view of the area is made the primary view.

Camera Selection Behavior / Focus-in-context View	Manual	Nudging	Automatic
No view			
Exo-centric	●		
Ego-centric	●		

Figure 12: Position of exo- and egocentric views in the design space.

4 Future Work and Discussion

Most important to us, we plan to conduct a controlled study on the effects of camera selection techniques and focus-in-context views on multi-camera remote collaboration. We will compare alternative versions of SceneCam on collaboration performance and user satisfaction. The conditions in the study are:

1. Manual camera selection with no focus-in-context view (core).
2. An exo-centric focus-in-context view with manual camera selection (exo).
3. Automatic camera selection with no focus-in-context view (auto).
4. Automatic camera selection with exo-centric focus-in-context view (auto+exo).

We will have pairs of participants (one local user and one remote user) go through an assembly task in a task space with multiple areas, where subtasks in one area are interdependent on subtasks in another area. We make the following hypotheses. Participants will perform better (faster completion time and fewer errors) in the exo, auto and auto+exo-conditions than in the core-condition. Participants perform better in the exo-condition than in the core-condition, because a remote user more easily can use the exocentric focus-in-context view to select an appropriate camera view based on the local user's projected path or descriptions of relative position ("look at the area to the left of me"), and more easily can point the local user to areas using descriptions of relative position ("go to the area behind you"). Participants perform better in the auto-condition than in the core-condition, because a collaborating pair spends less time negotiating the camera view, and the remote user has to spend little to no time on the meta-task of selecting the optimal view of the local user's actions. Participants perform better in the auto+exo condition than in all other conditions, because it combines the best of both worlds from the auto and exo-conditions.

One important limitation of the current SceneCam prototype is the accuracy (or lack thereof) of the 3D interpretation of 2D annotations. This inaccuracy may lead to misunderstandings between the

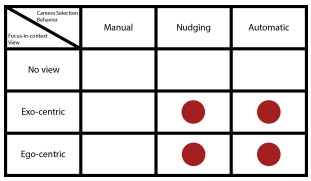

Figure 13: Combinations of camera selection techniques and focus-in-context views in the design space.

local user and remote user, for instance the local user may see the remote user point to an object which is different from the object the remote user is actually pointing to. Five factors contribute to the (in)accuracy of the 3D interpretation of 2D annotations:

1. Accuracy of the pose of the virtual camera.
2. Distance between real camera and task space.
3. Accuracy of the intrinsics of the real camera.
4. Accuracy of the inside-out tracking of the AR device.
5. Accuracy of the surface reconstruction of the AR device.

The virtual camera has a slight tendency to drift from the pose of the real scene camera. In the future we imagine using an intersection technique, such a natural hand gestures, to quickly re-align the virtual camera to the real one. Besides, we can improve the accuracy with which we acquire the intrinsics of the scene cameras by using a more systematic calibration procedure than is currently the case. The accuracy of the inside-out tracking and the surface reconstruction of the AR device is to a large extent outside our control as AR application designers and developers. Another limitation is the simplicity of the context-aware camera selection algorithm. The current algorithm uses "if-else" rules to infer the optimal camera view of the local user's actions. The context-aware camera selection technique, while an indispensable component that nudging and automatic camera selection depends upon, is not the sole focus of the paper, and thus a decision was made to keep it simple. It thus makes sense to conduct a separate study on different context-aware algorithms comparing how accurate they are at identifying the optimal camera view. It is desirable to be able to evaluate the accuracy of a context-aware camera selection algorithm, because the efficiency of the automatic camera selection and nudging techniques depends on it. An imprecise context-aware algorithm that produces many false positives, i.e. passes on the wrong camera view to the automatic camera selection or nudging techniques, will not be efficient, because it will force the remote user to spend time on manually undoing the camera selection.

5 Conclusion

In this paper we have presented SceneCam, a multi-camera AR remote collaboration prototype. To address some of the known usability issues of using multiple cameras for remote collaboration on physical tasks, we have presented different techniques and combinations thereof in a design space: nudging of camera selection, automatic camera selection and exo- and egocentric focus-in-context views. These techniques rely on the tracking capabilities of an AR device worn or held by the local user. We hypothesize that the techniques will improve usability of multi-camera remote collaboration by making it easier and faster to select the right camera view for a given collaborative task and plan to test our hypotheses in a controlled experiment in future work.

References

[1] H. Chen, A. S. Lee, M. Swift, and J. C. Tang. 3d Collaboration Method over HoloLens and Skype End Points. In *Proceedings of the 3rd International Workshop on Immersive Media Experiences*, ImmersiveME '15, pp. 27–30. ACM, New York, NY, USA, 2015. doi: 10.1145/2814347.2814350

[2] O. Fakourfar, K. Ta, R. Tang, S. Bateman, and A. Tang. Stabilized Annotations for Mobile Remote Assistance. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pp. 1548–1560. ACM, New York, NY, USA, 2016. doi: 10.1145/2858036.2858171

[3] S. R. Fussell, L. D. Setlock, and R. E. Kraut. Effects of Head-mounted and Scene-oriented Video Systems on Remote Collaboration on Physical Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pp. 513–520. ACM, New York, NY, USA, 2003. doi: 10.1145/642611.642701

[4] L. Gao, H. Bai, R. Lindeman, and M. Billinghurst. Static Local Environment Capturing and Sharing for MR Remote Collaboration. In *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*, SA '17, pp. 17:1–17:6. ACM, New York, NY, USA, 2017. doi: 10.1145/3132787.3139204

[5] W. W. Gaver, A. Sellen, C. Heath, and P. Luff. One is Not Enough: Multiple Views in a Media Space. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pp. 335–341. ACM, New York, NY, USA, 1993. doi: 10.1145/169059.169268

[6] R. Grasset, P. Lamb, and M. Billinghurst. Evaluation of Mixed-Space Collaboration. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '05, pp. 90–99. IEEE Computer Society, Washington, DC, USA, 2005. doi: 10.1109/ISMAR.2005.30

[7] C. Heath, P. Luff, and A. Sellen. Reconsidering the Virtual Workplace: Flexible Support for Collaborative Activity. In *Proceedings of the Fourth Conference on European Conference on Computer-Supported Cooperative Work*, ECSCW'95, pp. 83–99. Kluwer Academic Publishers, Norwell, MA, USA, 1995. event-place: Stockholm, Sweden.

[8] S. Kim, M. Billinghurst, and G. Lee. The Effect of Collaboration Styles and View Independence on Video-Mediated Remote Collaboration. *Comput. Supported Coop. Work*, 27(3-6):569–607, Dec. 2018. doi: 10.1007/s10606-018-9324-2

[9] J. Norris, H. Schndelbach, and G. Qiu. CamBlend: An Object Focused Collaboration Tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 627–636. ACM, New York, NY, USA, 2012. doi: 10.1145/2207676.2207765

[10] B. Nuernberger, K. Lien, T. Hllerer, and M. Turk. Interpreting 2d gesture annotations in 3d augmented reality. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 149–158, Mar. 2016. doi: 10.1109/3DUI.2016.7460046

[11] I. Poupyrev, S. Weghorst, M. Billinghurst, and T. Ichikawa. Egocentric Object Manipulation in Virtual Environments : Empirical Evaluation of Interaction Techniques. 1998.

[12] A. Ranjan, J. P. Birnholtz, and R. Balakrishnan. Dynamic Shared Visual Spaces: Experimenting with Automatic Camera Control in a Remote Repair Task. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pp. 1177–1186. ACM, New York, NY, USA, 2007. doi: 10.1145/1240624.1240802

[13] A. Stafford, W. Piekarski, and B. H. Thomas. Implementation of god-like interaction techniques for supporting collaboration between outdoor AR and indoor tabletop users. In *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 165–172, Oct. 2006. doi: 10.1109/ISMAR.2006.297809

[14] M. Tait and M. Billinghurst. The Effect of View Independence in a Collaborative AR System. *Computer Supported Cooperative Work (CSCW)*, 24(6):563–589, Dec. 2015. doi: 10.1007/s10606-015-9231-8

[15] F. Tecchia, L. Alem, and W. Huang. 3d Helping Hands: A Gesture Based MR System for Remote Collaboration. In *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, VRCAI '12, pp. 323–328. ACM, New York, NY, USA, 2012. doi: 10.1145/2407516.2407590

[16] K. Yamaashi, J. R. Cooperstock, T. Narine, and W. Buxton. Beating the Limitations of Camera-monitor Mediated Telepresence with Extra Eyes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '96, pp. 50–57. ACM, New York, NY, USA, 1996. doi: 10.1145/238386.238402