Online Real-Time Presentation of Virtual Experiences for External Viewers

Submission #130

ABSTRACT

Externally observing the experience of a participant in a virtual environment is generally accomplished by viewing an egocentric perspective. Monitoring this view can often be difficult for others to watch due to unwanted camera motions that appear unnatural and unmotivated. We present a novel method for reducing the unnaturalness of these camera motions by minimizing camera movement while maintaining the context of the participant's observations. For each time-step, we compare the parts of the scene viewed by the virtual participant to the parts of the scene viewed by the camera. Based on the similarity of these two viewpoints we next determine how the camera should be adjusted. We present two means of adjustment, one which continuously adjusts the camera and a second which attempts to stop camera movement when possible. We are able to show that our method can produce paths that have substantially shorter travel distances, are easier to watch and maintain the original observations of the participant's virtual experience.

Categories and Subject Descriptors

H.5.1 [**Information Presentation**]: Multimedia Information Systems—*Artificial, augmented, and virtual realities*

Keywords

Virtual Reality, Viewpoint Similarity, Camera Motion, Experimentation, Stabilization

1. INTRODUCTION

Monitoring the experiences of a user in a virtual environment is generally useful for spectators, scientists, architects, and designers. While some virtual reality hardwares allow external viewers to see both the individual and the projection, enclosed environments such as a C6 CAVE or a HMD provide little means to ascertain insight into the participant's virtual experience. In these cases, the traditional means in which viewers can gain an understanding of the user's experience is through the replication of the user's egocentric perspective. Unfortunately, this egocentric view is often difficult to watch as it is filled with movements that feel unnatural and unmotivated to an outside viewer [15].

Figure 1: External viewers often find it difficult to observe the experience of a virtual participant due to the unnaturalness of the egocentic camera movements. Our method is able to remove these unnatural motions by minimizing camera movement while maintaining the context of the participant's observations.

In this paper, we introduce methods for creating camera views for external viewers based on the experience of a participant in a virtual environment. The methods operate online in real-time, creating a camera path that meets two goals: conveying what the participant is seeing and providing camera movement that is easy to watch for external viewers. These two goals are often in conflict as the participant's control can lead to movements that they can anticipate (since they control them), but will appear as unnatural, unmotivated, and jerky to an outside viewer. Therefore, methods for creating views for external viewers must balance between being faithful to the participant's path, which fully conveys what they are seeing but may be difficult to watch, and simplifying the camera path to create an easy to watch video, at the expense of not conveying the participant's full view information.

Our approach to external view synthesis balances fidelity (conveying what the participant sees) with watchability. This is accomplished by using a content-dependent metric that limits the amount of motion of the camera based on how similar what it sees is with what the participant's camera sees. This effectively removes camera motions that are not necessary to show what the participant is seeing. Unlike a pure filter-based approach, our method will produce sharp camera motions if they are necessary for conveying content. The method operates on a per-frame basis, moving the camera towards the participant's view at each time step. The amount that

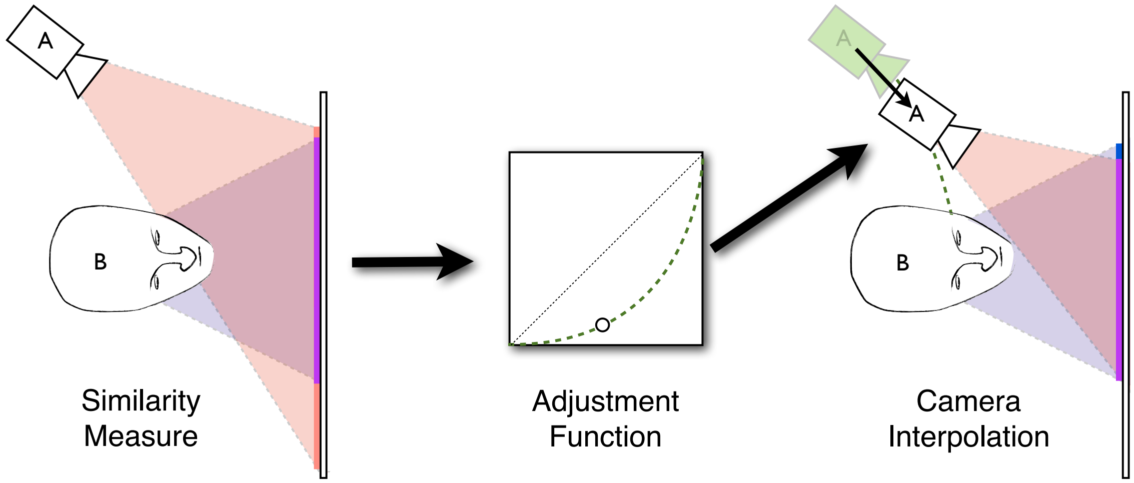

Figure 2: The system consists of three components for each point in time. First the similarity between the camera and the participant's viewpoint is determined. From this, an adjustment function is computed to determine the amount the camera should be repositioned. Finally the camera is moved in order to achieve a better view by interpolating it towards the participant's viewpoint.

the camera is allowed to move in each step is controlled by the view content similarity metric.

The contribution of this paper is a method for creating a watchable external view that conveys the experience of a virtual environment participant, online and in real-time. While the problem of effective external view creation has been considered in prior work, our approach is the first that operates online in real-time, and thus can serve important applications where observation occurs during the participant's experience. Our method builds on a prior view-dependent metric to provide an adaptivity that would not be possible with simple filtering. We demonstrate the effectiveness of our approach through quantitative experiments and a participant study.

2. RELATED WORK

Our work is inspired by recent results in video stabilization that show that optimization can explicitly balance between motion smoothness and faithfulness to the original camera [7, 8]. Unlike our work, these approaches use off-line optimizations and focus on challenges unique to 2D video stabilization.

Techniques to stabilize imagery in an online real-time fashion have been used in a variety of fields. Hansen et al. developed methods to stabilize satellite imagery in order to generate a larger photo mosaic [9]. Many researchers have used filtering techniques to stabilize video sequences in real-time [2, 5, 17, 20]. This type of filtering has become a particular focus for researchers dealing with mobile platforms [11, 14]. Others have used real-time stabilization as a means of treatment. For example, Pothier et al. used real-time image stabilization and augmented reality eyewear in the treatment of Oscillopsia for patients with bilateral vestibular loss [16]. Unfortunately, these types of image stabilization techniques generally do not handle the incredibly high-frequency movements of egocentric data gracefully [15].

Sensor fusion along with Kalman filtering has proven to be extremely useful to help stabilize the view of the participant in a virtual environment [6, 23]. These techniques are now common for commercial grade tracking equipment [13]. This type of filtering is unfortunately incredibly dependent on the conditions of the environment [5] and may therefore require parameters to be tuned for every different environment in order to be effective.

Researchers have studied changing the virtual perspective as a means to present different views of a virtual environment. Salamin et al. studied whether a participant inside of a Virtual and Augmented Reality situation would prefer to see themselves from a 3^{rd} or 1^{st} person perspective [18]. The authors found that the preferred perspective was very dependent on the task the participant was asked to accomplish. Yang and Olson tested whether 3^{rd} party commanders were able to direct more efficiently with either ego- or exo-centric viewpoints [22]. Their results also showed very mixed results, with ego-, exo-, or tailing cameras performing better depending on the situation. As the viewpoints were not being generated from a tracking device, no stabilization was needed.

The major motivation of our work comes from Ponto et al. who were able to show that egocentric viewpoints could be processed in order to create replay videos that were more effective at communicating participants' experiences in virtual environments for outside observers [15]. The authors' method decoupled the egocentric viewpoint from the replay camera path in order to minimize the camera movement and more effectively place the camera. Salient camera positions were determined using clustering by comparing viewpoints using a content dependent Viewpoint Similarty metric. The authors demonstrated that their methods were able to outperform simple filtering techniques in terms of their ability to convey content. Due to the offline nature of their algorithms, the techniques could not be implemented for an online application where future data are unavailable. We use the authors' similarity metric and presentation style as motivation for the development of our own online real-time algorithms.

3. OUR METHOD

Our basic idea is to produce a novel camera path in an online real-time manner by adjusting the camera only when necessary to maintain the content of the participant's observations. To do this, our method consists of three components as shown in Figure 2. First we determine the similarity between the viewpoints of the participant and viewer. From this similarity, we determine how much viewer's viewpoint should be adjusted. Finally the camera is repositioned by interpolating its previous position with the current participant's viewpoint.

3.1 Viewpoint Similarity

We chose to use the Viewpoint Similarity metric, described by Ponto et al. [15], as it provides a means to rapidly compare the content seen by two different viewpoints using a flashlight analogy. The method is based on a variant of shadow mapping [21], thus providing an efficient means for implementation on the GPU.

The method first computes the visibility of one viewpoint (B) in the context of the other viewpoint (A) as shown in Equation (1).

$$V(A,B) = \sum_{p=0}^{N} \frac{L(A_p,B_p)}{N}, \tag{1}$$

where $L(A_p,B_p)$ is a visibility function for a pixel, p of B in A and N is total number of pixels in the view. This function can be easily implemented on the GPU using occlusion queries [15]. As $V(A,B)$ is not equivalent to $V(B,A)$, the Viewpoint Similarity metric between A and B is defined as a weighted sum of the two viewpoints' relative visibilities:

$$S(A,B) = w_A V(A,B) + w_B V(B,A). \tag{2}$$

We use this computation to determine similarity between the viewpoints of the participant and the viewer. From Equation (2), we set A to the participant's viewpoint and B to the current state of the viewer's viewpoint, with the weights according to the scheme shown in [15]. This similarity is used to adjust the camera position as shown below.

3.2 Adjustment Function

After computing the Viewpoint Similarity, we create an adjustment function which indicates how much the camera's viewpoint should be altered. If the two viewpoints are highly similar, we can assume that camera needs very little adjustment. Conversely, if the two viewpoints are highly dissimilar, the camera viewpoint will need to be greatly repositioned. For our purposes, we have created two adjustment profiles, the Continuous Adjustment Profile (CAP) and the Stopping Adjustment Profile (SAP).

The CAP is designed to prioritize maintaining a camera position that is able to convey the content that the participant sees over easing the camera movement. For the CAP, we define u, the interpolation parameter as

$$u = k(1 - S(A,B)). \tag{3}$$

For this equation, the variable k defines a sensitivity parameter, which is adjustable by the participant. This value determines the maximum step size for each time step, meaning if two viewpoints are entirely dissimilar, this would be the max amount the camera should be adjusted. In practice we found the inverse of the rendering frame-rate to be a useful default value.

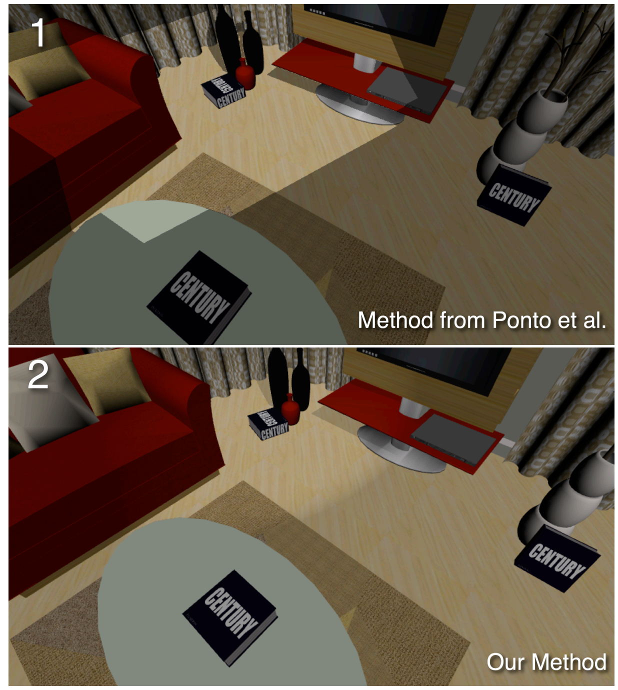

Figure 3: (Top) The rectangular light used by Ponto et al. with sharp edges of the light was shown to be distracting for viewers when both the camera and light were moving. (Bottom) Our method using a gradual falloff to provide probabilistic information as to where the participant was looking proved to be less distracting for the same situation.

While this first function constantly adjusted the camera position to increase the view similarity, others have shown that is useful to stop the camera movement when possible to help external viewers understand the virtual experience more effectively. Therefore, we introduce another profile, SAP, which prioritizes generating easy to watch camera movement over maintaining the precise view that the participant is seeing. For SAP, the interpolation parameter u, is defined as:

$$u = nk(1 - S(A,B))^n. \tag{4}$$

As the similarity value is normalized between zero and one, raising it to a large power will assure that highly dissimilar values will still adjust the camera, while marginally dissimilar values will not. As this function decreases camera movement, we also multiply the function by the exponent to ensure proper sensitivity.

In practice, both profiles have their advantages and disadvantages. In Section 4.2 we evaluate both methods to inform the choice between them and the selection of their parameters

3.3 Camera Interpolation

After determining the adjustment amount, we move the camera by interpolating between its previous position and the participant's viewpoint.

We adopt exponential maps for performing these interpolations as they rely on directly manipulating the viewing transforms, as op-

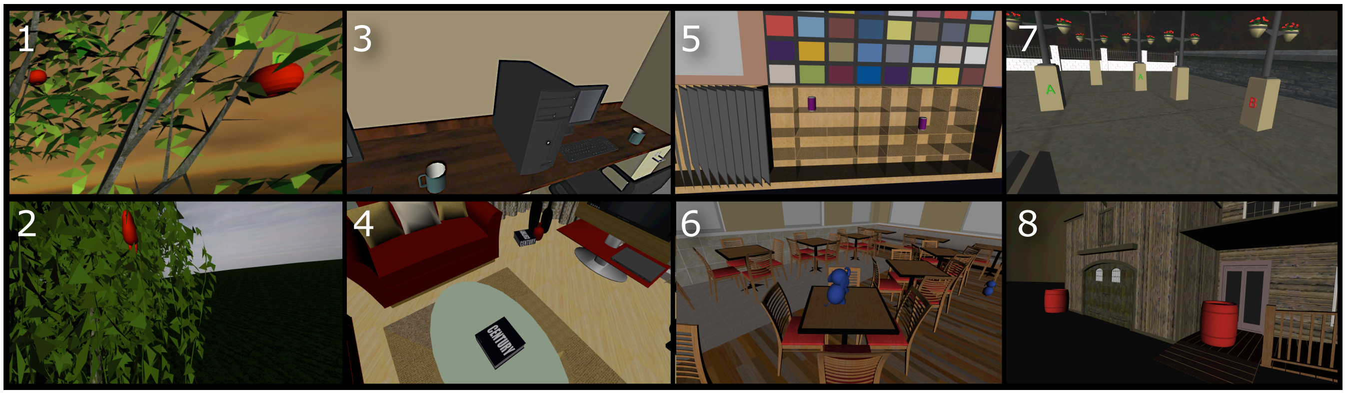

Figure 4: Images taken from the eight different scenarios for the participant to locate virtual objects.

posed to interpolating the various camera components independently. Exponential maps accomplish this by performing linear interpolation operations in the logarithm space of the matrices [1]. We chose to use exponential maps as opposed to parameter interpolation as, in common cases, they do a better job of keeping points of interest centered in view, as shown by Hawkins and Grimm [10]. It is worth noting that as the interpolated amounts are often quite small, the difference between parameter interpolation and exponential maps is likely negligible. After interpolation we remove camera roll to produce views that are level in the horizontal axis, following the practices of Ponto etl al. [15].

Equation (5) demonstrates how $A(t)$, the next position of the viewer's viewpoint, is calculated given the viewpoint of the viewer ($A(t-\Delta t)$) updated previously, the participant's viewpoint ($B(t)$), and the adjustment parameter (u) determined from Section 3.2.

$$A(t) = e^{(1-u)\log A(t-\Delta t) + u \log B(t)}. \quad (5)$$

While this method does not have any explicit easing parameters as were shown in [7] and [15], some smoothing will happen naturally assuming that the viewpoints do not change too rapidly. For instance, as the participant slowly changes their view, the Viewpoint Similarity will also decrease, thereby increasing the interpolated step-size. Conversely, as the the viewpoints become more similar, the Viewpoint Similarity metric will also increase, thus decreasing the interpolated step-size.

3.4 Presentation

As shown by Ponto et al., manipulating a camera path may reduce an external viewer's understanding of the participant's experience [15]. To reduce this artifact, the authors drew a rectangular light representing the participant's viewpoint as shown in Figure 3-1 to provide the viewer with a clear representation of where the participant is looking. However, this rectangular light with a sharp cutoff was shown to be distracting when the light and the camera were moving at the same time. Furthermore, the field of view represented by the rectangle did not precisely match that of the participant's perspective. We instead created a virtual light with a gradual fade-off to provide probabilistic information as to where the participant was looking, as shown in Figure 3-2. This proved to also be less distracting for the viewer when the camera was in motion.

4. RESULTS

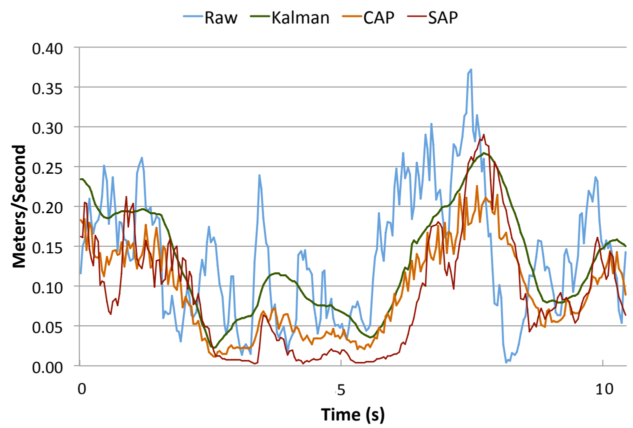

Figure 5: Velocity profile for approximately 10 seconds of a participant walking around a virtual tree (Scene 1). As shown, our method reduces the magnitude of the velocity compared to the Raw or Kalman filtered camera path.

The goal of our method was to produce new camera paths in an online real-time fashion that reduce the camera movement, thus making them easier to watch without obscuring content from the raw camera path. In order to determine if our methods achieved these objectives, we first generated camera paths by asking participants to walk around in a fully enclosed C6 CAVE and find a particular type of objects within one of several virtual environments.

These eight scenes are shown in Figure 4. The scenes could be grouped into four classes: small scale walk-around, small interior environments, large interior environments, and larger outdoor spaces. In the first group, the participants were tasked with counting apples (Figure 4-1) and birds in a dense bush (Figure 4-2). The small interior environments included an office with coffee mugs (Figure 4-3) and a living room with books (Figure 4-4). In the two larger interior spaces, the participant was tasked with locating toys in a small restaurant (Figure 4-5) and cans in a store (Figure 4-6). For the last group, we created a park scene with light posts (Figure 4-7) and an old western town with barrels (Figure 4-8).

During these tasks, participants' head positions and orientations were recorded for further analysis. To assess the relative performance of our method to other approaches, we considered four dif-

Camera Travel Distance [m] (Distance / Raw)				
Scene	Raw	Kalman	CAP	SAP
1	5.0	4.4 (.88)	3.2 (.64)	2.8 (.56)
2	14.3	12.6 (.88)	11.2 (.78)	9.4 (.66)
3	9.8	9.1 (.93)	6.1 (.62)	4.5 (.46)
4	11.3	11.3 (1.0)	9.3 (.82)	7.3 (.65)
5	31.1	31.0 (1.0)	26.5 (.85)	25.8 (.83)
6	27.9	30.3 (1.1)	25 (.90)	24.1 (.86)
7	33.5	33.5 (1.0)	29.5 (.88)	26.9 (.80)
8	90.1	100.1 (1.1)	85.0 (.94)	85.7 (.95)
Average Ratio		.99	.81	.72

Table 1: Camera movement metrics for 30 seconds of each scene for each method. Note that while the Kalman filter has very mixed results in reducing camera travel distance, our methods are able to consistently reduce the distance the camera traveled.

ferent on-line methods.

The first path consisted of the raw egocentric viewpoint, which is used for external viewers in most practical implementations. For the second path, we adopted a Kalman filter with constant velocity models for camera position and orientation [19]. The filter parameters including measurement errors and process errors, were set to be enough to suppress high-frequency components in raw camera movements while preserving its important features without significant delay. The third path was constructed using the CAP method (Equation (3)) with k set at 0.016. The fourth path was constructed using SAP method (Equation (4)) with k set at 0.016 and n set at 64. Each of the four methods was applied to the traces from all of the scenes.

4.1 Performance

While the Kalman filter requires very little computation overhead to process, the CAP and SAP methods due incur some computational penalties. To compute the Viewpoint Similarity, the scene must be rendered from both the camera and the participant positions, thus requiring a second render of the geometric models. Secondly, the Viewpoint Similarity metric requires a screen-space computation to be completed for each comparison (Equation (2)). Finally, the flashlight overlay (Figure 3) requires one final screen-space pass. We rendered each of the eight scenes into a 960x540 sized window on a computer equipped with an Intel Xeon 2.67 GHz CPU, 24 GB RAM, and an Nvidia Quadro 5000 GPU. We were able to achieve a rendering rate greater than 60 Hz for both the CAP and SAP method.

The total path length traveled by the camera provides an accessible, quantitative assessment of camera motion. It has been shown that simpler and shorter camera paths are desirable because they are easier to watch [15]. To test whether the CAP and SAP methods did reduce the camera movement, we monitored the change in camera position at each time step for each of the methods. Figure 5 shows an example velocity profile demonstrating that both the CAP and SAP methods tend to have a lesser velocity compared to Raw or Kalman filtered camera paths.

Table 1 shows the distance the camera traveled for each of the methods, as well as its ratio compared to the original camera path. As shown, the Kalman filter does not affect the amount the camera travels compared to the Raw path on average for the listed experiences. This makes sense as the Kalman filter is targeted to remove the high-frequency components of the camera movement and thus can overshoot the targeted camera motions. The CAP and SAP methods both reduce the distance the camera traveled in all eight scenes, with the SAP method able to reduce the camera motion by over 50% for some scenes.

4.2 Study

We designed a study to test whether the CAP and SAP methods were able to create a better viewing experience without obscuring information. Our hypothesis was that viewers should be able to perform the same counting tasks asked of the participant in the virtual environment based solely on the visual images being shown to them. We hypothesized that our method could improve the viewing experience for our study subjects, allowing them to complete the counting tasks more easily and with more confidence, and without reducing their performance compared to viewing the raw camera path.

We ran the experiment through Amazon mechanical Turk, following the practices described by Kittur et al. [12] and Downs et al. [4]. Specifically, we chose a standard rate of pay and created questions to verify that the participant was human, had the technical ability to participate, and was actively engaged. Subjects were compensated based on an estimate that the experiment would require approximately 15 minutes of their time.

After completing questions about demographic information, the subjects were shown eight videos corresponding to a certain scene and method. The order of the videos and methods was randomly selected so that each subject would see each scene once and each method twice. The study subjects were asked to count the same objects that the participant was asked to count inside of the virtual environment. After watching each video, the participant was asked to input their counted value and input a prominent floating word from the video as an engagement check. The participant was then asked a series of questions about the camera movement, the difficulty of the task, and their confidence in their answers. Each question was answered on a five point Likert scale. The study attracted 24 subjects, 14 males and 10 females from ages 18-69. All subject's data were used.

We found that subjects were generally able to count the objects accurately, with subjects able to produce the correct result 57% of the time. We found no significant difference in the error of the object count between the methods. We also found no significance between the error count and the scene being shown. The result confirms our hypothesis that subjects can complete the same counting task and that the CAP and SAP methods did not obscure the content of the original observations.

Figure 6 shows graphs for each of the subjective questions asked in the study. We note that while these questions used a five-point Likert scale, research has shown Likert scales to be an approximate of a continuum [3]. As shown, the CAP and SAP methods were able to perform well in all of the categories listed.

We found that subjects rated the calmness of the camera movement to be statistically significant, $F(3, 188) = 18.105$, with Tukey post-hoc analysis showing the CAP and SAP methods to be significantly better than the Kalman and Raw methods. Subjects' ratings of the smoothness of the camera movement also proved to be statistically

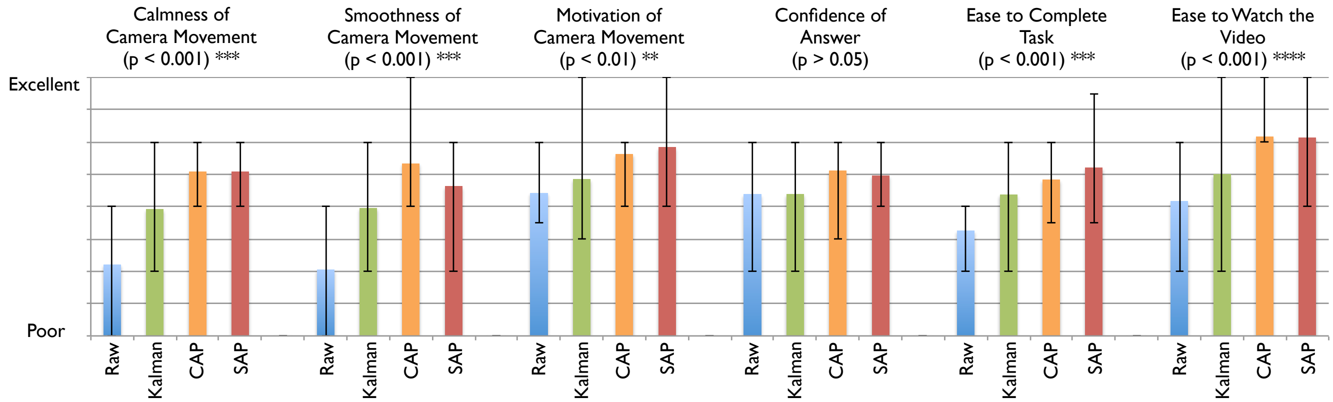

Figure 6: Graphs for each of the subjective questions asked in the study. The results proved to be statistically significant, with the only exception for the subjects' confidence in their answers.

significant, F(3,188) = 19.385, with Tukey post-hoc analysis showing the CAP and SAP methods to be significantly better than the Raw method and the CAP method to be significantly better than the Kalman method. Subjects' ratings of the motivation of camera movement proved to be statistically significant, F(3,187) = 4.4988, with Tukey post-hoc analysis showing the CAP and SAP methods to be significantly better than the Raw method.

We also found that subjects rated the ease with which they could accomplish the counting task to be statistically significant, F(3,188) = 5.9002, with Tukey post-hoc analysis showing the CAP and SAP methods to be significantly better than the Raw method. Finally, we found subjects' ratings of the ease to watch the video were statistically significant, F(3,188) = 7.1315, with Tukey post-hoc analysis showing the CAP and SAP methods to be significantly better than the Raw method. We did not find a significant difference for subjects' confidence for answering the questions about the number of objects in a scene. These results confirmed our hypothesis that viewers would find the CAP and SAP methods easier to watch and find the task easier to accomplish. While we did see a higher confidence for answering the questions with the CAP and SAP methods, the result was not statistically significant.

5. DISCUSSION

From the results of our study, we feel that both the CAP and SAP methods are superior to the standard method of presenting views using raw egocentric camera motion. The SAP method slightly out-performed the CAP method in most metrics with the exception of the smoothness test, which is to be expected. From this result it would be interesting to study a variety of settings for the exponential parameter, n, in the SAP method.

The results of our study show that the CAP and SAP methods perform equally well in comparison to the Raw path in terms of the counting of objects, thus indicating that our method does not obscure the content of the original path. While it was outside of the scope of this paper, it would be interesting to determine if any method would be able to outperform the original observations. Although we did not see any significant difference between the methods, we think it is worth mentioning that the variance of the normalized object count is smaller with the SAP method (0.010) compared to the Raw, Kalman and CAP methods (0.032, 0.026, and 0.020 respectively).

As shown in Figure 6, the subjects were not significantly more confident in their answers with the path from either CAP or SAP method than the others. However, we found no correlation between the subject's confidence in their answer and the correctness, suggesting that self-reported confidence is not a good predictor of performance for this task.

Another interesting finding is that geometric smoothness does not always coincide with perceptual smoothness. While the velocity profile of the Kalman filter appears to be smoother than either the CAP or SAP methods, the subjects rated the camera movements of both CAP and SAP methods to be more calm and/or smooth.

For all results of our paper we used a GPU implementation that utilized an offscreen buffer with the same sized window. However, this may be undesirable in situations with extremely complicated scenes, ultra-high-resolution displays, or lower-end graphical hardware. By reducing the size of the buffers used to store the depth maps for the participant's view and the camera's view, both the scene rendering and pixel-wise comparisons can be done more rapidly. As shown in Ponto et al., reducing the size of these buffers can greatly increase performance without affecting results [15].

6. CONCLUSIONS

We created a novel method for presenting the experience of a participant in a virtual environment to external viewers in an online real-time fashion. Our method uses a content dependent metric to determine how much the camera should be adjusted on a per-frame basis. Our method is able to greatly reduce the camera movement over time and is thusly able to provide camera paths that are easier and more pleasing to watch while maintaining the content of the participant's original observations. Future work will explore using other viewing perspectives, such as an over the shoulder viewpoint, and will investigate methods to change the camera's field of view dynamically.

7. REFERENCES

[1] M. Alexa. Linear combination of transformations. *ACM Trans. Graph.*, 21:380–387, July 2002.

[2] H.-C. Chang, S.-H. Lai, and K.-R. Lu. A robust real-time video stabilization algorithm. *Journal of Visual Communication and Image Representation*, 17(3):659 – 673, 2006.

[3] J. Dawes. Do data characteristics change according to the number of scale points used? *International Journal of Market Research*, 50(1), 2008.

[4] J. S. Downs, M. B. Holbrook, S. Sheng, and L. F. Cranor. Are your participants gaming the system?: screening mechanical turk workers. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 2399–2402, New York, NY, USA, 2010. ACM.

[5] S. Ertürk. Real-time digital image stabilization using kalman filters. *Real-Time Imaging*, 8(4):317 – 328, 2002.

[6] E. Foxlin. Inertial head-tracker sensor fusion by a complementary separate-bias kalman filter. In *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996*, pages 185 –194, 267, mar-3 apr 1996.

[7] M. L. Gleicher and F. Liu. Re-cinematography: Improving the camerawork of casual video. *ACM Trans. Multimedia Comput. Commun. Appl.*, 5:2:1–2:28, October 2008.

[8] M. Grundmann, V. Kwatra, and I. Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 225–232. IEEE, 2011.

[9] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P. Burt. Real-time scene stabilization and mosaic construction. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pages 54 –62, dec 1994.

[10] A. Hawkins and C. M. Grimm. Camera keyframing using linear interpolation of matrices. *journal of graphics, gpu, and game tools*, 12(3):55–69, 2007.

[11] Y. Kim, V. Jayanthi, and I. Kweon. System-on-chip solution of video stabilization for cmos image sensors in hand-held devices. *IEEE transactions on circuits and systems for video technology*, 21(10):1401–1414, 2011.

[12] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 453–456, New York, NY, USA, 2008. ACM.

[13] J. Liang, C. Shaw, and M. Green. On temporal-spatial realism in the virtual reality environment. In *Proceedings of the 4th annual ACM symposium on User interface software and technology*, UIST '91, pages 19–25, New York, NY, USA, 1991. ACM.

[14] C.-T. Lin, C.-T. Hong, and C.-T. Yang. Real-time digital image stabilization system using modified proportional integrated controller. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(3):427 –431, march 2009.

[15] K. Ponto, J. Kohlmann, and M. Gleicher. Effective replays and summarization of virtual experiences. *Visualization and Computer Graphics, IEEE Transactions on*, 18(4):607 –616, april 2012.

[16] D. D. Pothier, C. Hughes, W. Dillon, P. J. Ranalli, and J. A. Rutka. The use of real-time image stabilization and augmented reality eyewear in the treatment of oscillopsia. *Otolaryngology – Head and Neck Surgery*, 2012.

[17] K. Ratakonda. Real-time digital video stabilization for multi-media applications. In *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on*, volume 4, pages 69 –72 vol.4, may-3 jun 1998.

[18] P. Salamin, D. Thalmann, and F. Vexo. The benefits of third-person perspective in virtual and augmented reality? In *Proceedings of the ACM symposium on Virtual reality software and technology*, VRST '06, pages 27–30, New York, NY, USA, 2006. ACM.

[19] H. J. Shin, J. Lee, S. Y. Shin, and M. Gleicher. Computer puppetry: An importance-based approach. *ACM Trans. Graph.*, 20(2):67–94, Apr. 2001.

[20] C. Wang, J.-H. Kim, K.-Y. Byun, J. Ni, and S.-J. Ko. Robust digital image stabilization using the kalman filter. *Consumer Electronics, IEEE Transactions on*, 55(1):6 –14, february 2009.

[21] L. Williams. Casting curved shadows on curved surfaces. *SIGGRAPH Comput. Graph.*, 12:270–274, August 1978.

[22] H. Yang and G. M. Olson. Exploring collaborative navigation:: the effect of perspectives on group performance. In *Proceedings of the 4th international conference on Collaborative virtual environments*, CVE '02, pages 135–142, New York, NY, USA, 2002. ACM.

[23] S. You and U. Neumann. Fusion of vision and gyro tracking for robust augmented reality registration. In *Virtual Reality, 2001. Proceedings. IEEE*, pages 71 –78, march 2001.