Predictive Display from Computer Vision Models

Martin Jagersand, Adam Rachmielowski, David Lovi, Neil Birkbeck,
Alejandro Hernandez-Herdocia, Azad Shademan, Dana Cobzas, Keith Yerex
University of Alberta

Abstract— **In tele-manipulation, delays as small as a few tenths of a second can affect performance. Robot operators dissociate their control actions with what they see on delayed video and have to adopt slow move-and-wait strategies or may completely fail to perform high precision tasks. Predictive Display (PD) mitigates this problem by rendering visual feedback that reflects the operator's motion immediately. Conventional PD is based on a-priori CAD models and calibrations. Using modern computer vision, we have implemented on-line model capture and tracking which allows the rendering of textured graphical PD. We validate different types of PD and compare them to using delayed video in an alignment task. Graphical PD was found comparable to a no-delay situation, while task completion on average took 48% longer with a relatively short 300 ms delay.**

I. Introduction

In tele-manipulation, a human operator controls the motions of a remote robot manipulator [1]. Typically, the operator uses a high-DOF mechanical joystick (e.g. Phantom or similar), though other interfaces are possible (e.g. optical tracking). The slave robot replicates the motions at the remote site. The robot can include an arm, gripper or multi-finger hand. The operator views the manipulation scene through a remote video camera that can either be mounted on the arm or fixed. Despite usually using straightforward kinematics between the master and slave robot, tele-manipulation is surprisingly difficult and slow when compared to a human physically directly performing a task. A particular difficulty is the delayed visual response from the tele-manipulation system. Delays can be caused by distance, but other common causes include network switching delays, processing delays and slow dynamics of the slave manipulator. Additionally, the typically limited field of view of robot cameras and their limited and/or slow articulation compared to the human eye and eye movements may also deteriorate performance. Yet tele-operation is often the preferred or only feasible mode of operation. High-end applications include space and medical microsurgery. In space, sending human astronauts is expensive (near earth) or not yet feasible (planetary exploration), and autonomous robotics and AI have proven insufficient. In microsurgery, the scale of operation is so small that it is difficult for direct human manipulation, but a tele-robot can scale the magnitude of the motions.

To mitigate effects of delay and dynamics, the robot video feed can be augmented or replaced by rendered visual feedback. This rendering can be done using a system model which forward predicts in time. The rendering can show a wider field of view and a different viewpoint by using a 3D model and textures that have been integrated from many viewpoints acquired during the past motion of the robot and camera. The goal of a predictive display system is to provide the user with the feeling of being situated at the remote site and directly performing the manipulations. Just as the goal of a radio and its audio system is to replicate the performance of a musician as if the radio listener were present, this requires a degree of fidelity of predictive display systems. It has been shown that even very short delays between that of an operator making a motion and seeing the result severely affects human operator performance. When humans physically manipulate objects, motions are smooth and end-point positioning is guided by direct visual feedback. Experiments have shown that when a time delay is introduced in the visual feedback loop the manipulation performance degrades rapidly. Early experiments indicate that delays as short as 0.3 seconds break human hand-eye coordination [2]. Acceptable delay times depend on the type of task and system, e.g. the acceptable delay in a head mounted display (HMD) w.r.t. head motions is shorter (less than 0.1 s) than for arm motions.

Early work in tele-manipulation investigated how humans adapt to delay and its effect on task performance [3]. Using tele-robotics setups of the 1960's and choosing tasks to be doable by these, they found that humans adapted to delay by making small carefully judged motions and waiting for the delayed video to arrive after each motion. Thus task completion time increased linearly with delay time, accounting for the increased wait time. It is also reasonable to assume that a move-and-wait strategy incurs a higher cognitive load on the human than natural manipulation. Modern tele-manipulation masters are significantly more precise. Move-and-wait strategies may not work for high-DOF high-precision alignments. Intuitively, the reason is that while the human may try to cognitively adjust a particular freedom this will cause the others to drift out of alignment. Even holding steady a precise 6 DOF pose with a 6 DOF master (e.g. Phantom) is considerably harder with delayed feedback than with direct vision. Hence, with increased sophistication and precision of tele-manipulation systems, predictive display will be increasingly important.

Predictive display and (more generally) augmented display systems come in many varieties. Early systems were based on an a-priori graphics model and would render this as a wireframe on top of the delayed video [4]. Many systems require a fully calibrated system (robot, camera, object and scene coordinates)[5].

In the past two decades, the capability of computer vision

to acquire scene geometry and calibration has improved. Consequently, newer computer predictive display systems are able to acquire more information from the scene and rely less or not at all on a-priori models [6], [7], [8] This is obviously important in exploration tasks, where the mission is to obtain information of an unknown environment, but it can also be crucial in tasks such as on-orbit servicing, where in principle models are available, but in practice these may not be sufficient. For example, when servicing a mechanical failure, one cannot rely on the a-priori CAD model of the failed assembly since the mechanical assembly may have deformed or fallen apart. While servicing an electrical failure, it may be useful to inspect for signs of components having been burnt or connections corroded.

II. Types of Predictive Display

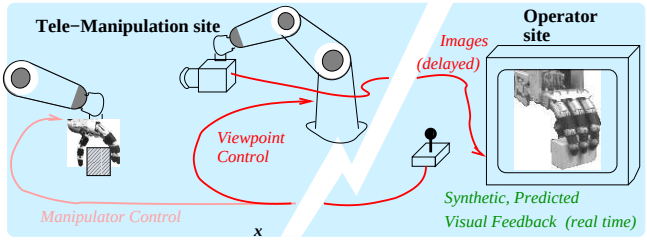


Fig. 1: Remote tele-manipulation setup.

Consider a tele-robotics setup as in Figure 1, where an operator controls a remote robot. The remote scene is viewed by a camera mounted on the robot. The scene images are shown to the operator using e.g. a video screen or head mounted display (HMD).

Let $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t, \ldots)$ be a sequence of viewpoint motion commands by the tele-operator. Assuming a round-trip delay d, the operator will not see the results of the current motion $\mathbf{x}_t$ until time $t + d$. In the experiments sections an electrical circuit board scenario will be used for inspection and motion alignment tasks. Fig. 2a illustrates the effect of delay in such a scenario. The operator has moved the robot to the desired viewpoint pose, and would expect to see the view from the desired camera looking straight down onto the circuit boards. Instead, due to delays and dynamics of the robot, he sees the previous view from the slanted camera to the right in the figure.

In graphical predictive display an estimated image $\hat{I}$ from the desired scene viewpoint $\mathbf{x}_t$ is rendered immediately from an image-based model M. The model is generated using a sequence of (previous) images from the remote scene $(I_1, I_2 \ldots I_m)$ as training data. The model consists of a viewing geometry describing cameras, robot and scene and a method for how to use the previous images to texture and render new views. Various robot manipulation setups lend themselves to using different model representations as will be detailed later.

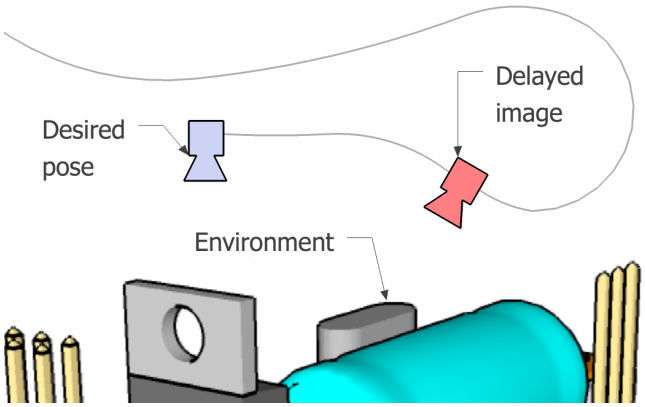
In terms of geometry, image-based-rendering (IBR) techniques relate the pixel-wise correspondence between sample images $I_{1\ldots m}$ and the synthesized desired new view $\hat{I}$. This can be formulated using a forward warp function w_t to compute the predictive display by forward warping the latest received remote image $\hat{I} = I_{t-d}(w_t)$. Alternatively several previous images can be combined to render a larger field of view.

Conceptually the warp function involves projecting input images on a 3D geometry, then rendering this geometry from a desired viewpoint. In practice these two operations are combined into one composite warp w, either directly from input images, or a texture space where all input images have been unified. Generally in computer vision and graphics, a very detailed 3D model and a non-linear perspective camera is used, so that a model consisting of a single geometry and texture can be rendered accurately from any viewpoint. Computing such a detailed model is a fragile and time consuming process and currently there is no system that reliably works in all environments. In predictive display sometimes only a moderate perturbation of the input image is needed. E.g. in the case of a moderate delay, $I_t \approx \hat{I}$, and then w is close to the identity function, and relatively insensitive of the underlying 3D geometry used. However, to relate arbitrary viewpoints, w can be quite complex.

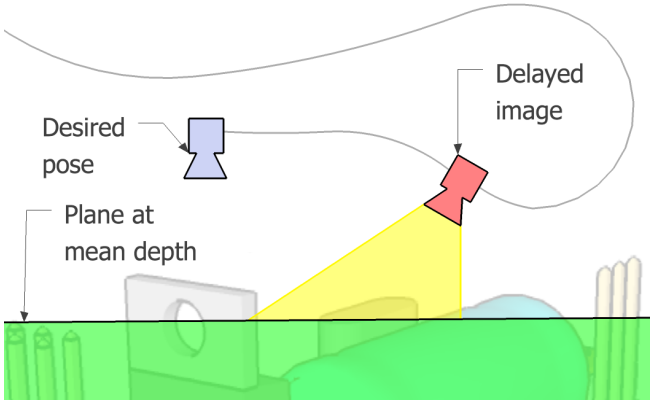
Tele-robotics is by definition on-line. Hence our system needs to be capable of rendering within sub seconds of receiving new information. A new situation faces the tele-robotics system at start-up in a new remote environment, and also when, for instance, the robot camera is moved to a new field of view for the first time, as well as when something changes in an unanticipated way in the scene.

Predictive display needs to quickly respond to new scene information by transition between image-based and model-based rendering, incorporating more 3D information on-line as it is being calculated. The following describes a hierarchy of increasingly richer models and display.

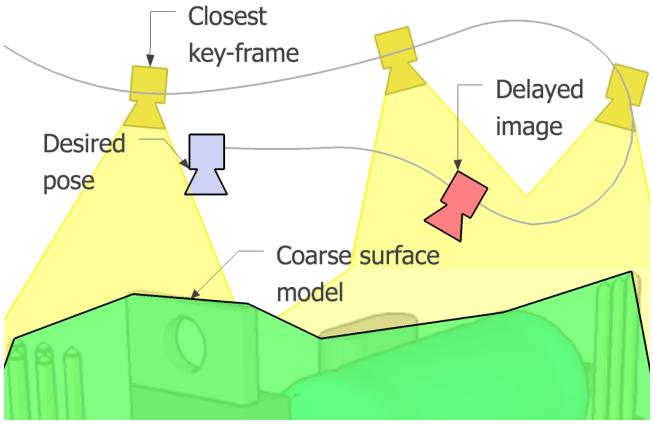
1) **2D image-based forward warp.** When only a small change in viewpoint is required, a planar warp can represent small robot camera motions. This technique is used in commercial camcorders by warping the image plane to stabilize video and remove the shaking in a handheld camera. While technically this is the correct transform only if the motion is either a camera rotation or the scene is planar, it works well in practice for general camera motions and scenes when the depth variation in the scene or camera translation is small. In the framework of a 3D geometry, we can represent the planar warp as an image transfer via a scene plane as illustrated in Fig. 2b. Such a plane can be placed at the expected or measured scene depth and made coplanar with the image-plane or to best align with the scene. In the former case the plane can for instance be placed at 80% of the robot arm reach and thus no external scene information is needed. Hence this type of prediction is directly available even in an unknown environment. As 3D scene points are estimated from the video, the plane can be adjusted to align with the actual scene. As more points are acquired, the plane can be broken into several facets, eventually forming a geometry as



(a) No delay: images are shown directly from the robot pose as specified by the user. Delayed image: delayed images are shown, which do not reflect the desired pose of the robot.



(b) Stabilizing plane PD: the delayed image is back projected onto a plane that is rendered from the desired pose.



(c) Model-based PD: the closest key-frames are back projected onto a coarse surface model and then rendered from the desired pose.

Fig. 2: Forward prediction with delayed and saved images.

described next. Note that a planar warp is uniquely specified by four points, and thus a plane is a four point projective model.

2) **Forward warping using a stabilizing 3D structure.** As the robot camera moves about the remote scenes, tracking and on-line modeling (detailed later, Section III) is used to estimate a set of 3D model points, on-line. Based on only a few dozen such model points, a better approximation of the true 3D change can be computed by forward projecting the scene video onto a triangulated surface defined by the 3D structure, and then back-projecting this into the desired virtual view for the operator. This is indicated in Fig. 2c when texturing only from the delayed (pink) video camera.
3) **3D model-based predictive display.** During operation, video is continuously acquired from the robot camera. A subset of this from suitably spaced viewpoints can be saved as key frames, and a larger field of view can be textured, see Fig. 2c where texturing is from the yellow keyframe cameras. Likewise after an extended time of operation a denser set of 3D points has been computed. We can then switch from forward warping video to representing both geometry and texture in a unified 3D model. Unlike the forward warping, this allows the rendering of images from any viewpoint, and now operator view point can be decoupled from the pose of the robot cameras. This is desirable e.g. in robot manipulation when the camera(s) are mounted on the arm, and the motions needed for robot manipulation do not necessarily give the best viewpoints.

While 3D model based predictive display may seem superior, it is not always necessary. As seen in the experiments, plane-based PD performs remarkably well and in comparison to delayed video. It is also worthwhile considering the accuracy of the 3D model points. A detailed model with many, but inaccurate 3D points is likely to introduce worse rendering artifacts than a plane model (or sparse, smooth model surface based on only accurate points).

III. Acquiring Models from Camera Video

Given images of a scene or object, the 3D geometry can be recovered in a variety of ways. Most however work on a batch of images, not incrementally, and are slow due to manual requirements (e.g. calibration and correspondences in photogrammetry) or CPU intensive (automatic systems relying on computational search for dense stereo, e.g. [9]). Practically care must be taken in selecting both scenes and viewpoints for the system to work well.

Luckily for predictive display the geometry can be quite coarse and approximate. Early graphics PD work relied on linear camera models [6]. This works well if the scene has limited extent, such as the workspace of a fixed manipulator. For mobile manipulators, a full non-linear viewing geometry model is needed. More recently real time systems for monocular SLAM [10] and SFM [11], [12] have become available. Mono-SLAM is an extension of Simultaneous Localization And Mapping from typically using direct depth

measurements and reconstructing a 2D floormap, to inferring depth from 2D images, and reconstructing a full 6D camera and 3D geometry. Real-time SFM (Structure-and-Motion where "Motion" refers to recovering the 6D camera pose) is an offshoot of earlier off-line methods, where time consuming steps such as the bundle adjustment has been divided into computational pieces small enough to run in the interframe intervals, see Fig. 3. We have used all of the above three referenced systems with success. For the results in the experiments section the third (PTAM) was used.

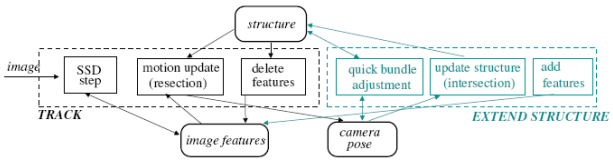


Fig. 3: Real-time SFM time-slicing

From SFM, we have 3D point structure, camera poses, and visibility information relating which points are visible from each keyframe camera pose. To construct a single consistent 3D model for rendering from a wider range of viewpoints, we exploit this visibility information in the form of constraints on the scene at the robot site. The notion of such a "free space" constraint is depicted in Fig. 4. Assuming scene opacity, we know that the volumes comprised of the rays of projection between viewed 3D features and their respective viewing cameras are empty. Thus these volumes can effectively be carved away. Since our 3D features are points, our free-space volumes are infinitesimally thin line segments, as depicted in the figure.

To carve a model, we naturally take a volumetric approach: we discretize space via the 3D Delaunay triangulation and carve away volume elements if they intersect a free-space constraint. The Delaunay triangulation chops space into a set of connected tetrahedral volume elements that span the 3D point set. The boundary between the carved and uncarved

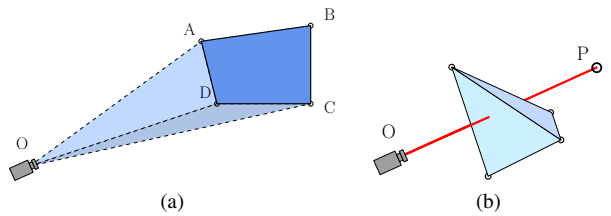


Fig. 4: Free-space constraints. (a) The general concept. A camera O observes a surface patch, here the quadrilateral $ABCD$. The pyramidal volume $ABCDO$ must be empty; otherwise, the patch would be occluded. (b) Our chosen representation of free-space constraints. The carving method considers only points P instead of generalized patches. Therefore our free-space constraints are the infinitesimally thin volumes, the line segments, $\overline{OP}$. We carve away inconsistent tetrahedra.

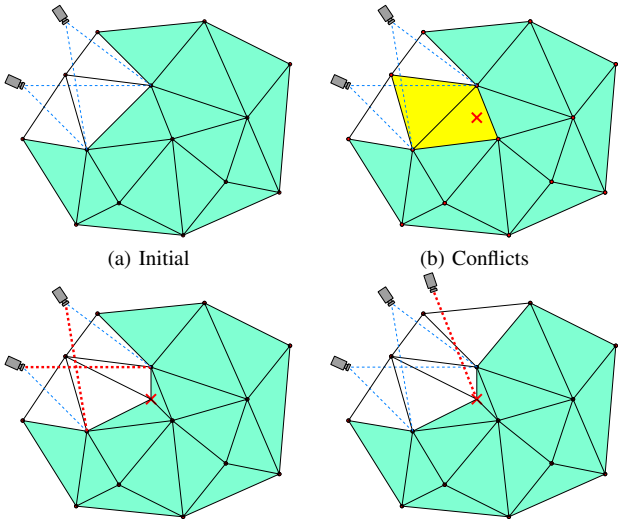


Fig. 5: A 2D illustration of incremental free-space carving. (a) The initial triangulation. The blue dashed lines are free-space constraints currently carving the triangulation. Shaded aquamarine cells have not yet been carved. (b) An incoming point (red cross). The yellow cells are in Delaunay-conflict with the point because it falls inside their circumcircles. (c) The yellow cells were deleted and rediscretized to account for the new point. The red free-space constraints (bolded) used to carve the deleted tetrahedra; they are now used to carve away two of the four new tetrahedra. (d) Finally, the new free-space constraint(s) from the current view are applied.

volumes is a set of triangular facets that defines the scene surface. We output this boundary as a conventional 3D graphics mesh for rendering.

For predictive display, fast online operation of such a 3D modeling method is a requirement. Our method benefits in speed from being fully incremental. As time passes and the robot camera acquires more and more video frames, online Structure and Motion continues to operate and produce more information. Therefore our modeling method's inputs, namely the point set, camera track and visibility information, continuously change online as they are refined and augmented. Instead of starting over from scratch and reprocessing the entire input after each such change (e.g., a new video frame or an outlier deletion), our method reconciles the carving to reflect the changes incrementally. We processes only the minimal set of information necessary, and we achieve real-time performance.

An illustration of the incremental algorithm's operation is provided in Fig. 5. The exact algorithmic details are outside the scope of this paper. However, a complete description, pseudo-code and modeling results can be found in [13], as well as complexity proofs and timings that highlight the real-time quality of our approach. In this paper, we show a sample reconstructed 3D model of our robotics lab in Fig. 9.

IV. System with On-line Texture Selection and Rendering

The predictive display system has to handle on-line updating to its various structures in real time, Fig. 6. Aspects of this that are directly related to the building of the 3D geometric model were discussed in the previous section. In addition, keframes for texturing has to be selected and stored in various places. A video frame is selected as a key frame if its computed calibration is accurate (95% confident), and it will provide a novel view. Novelty is numerically evaluated based on the mean distance to the scene and angular view difference compared to nearby cameras already in the keyframe set.

Images are stored in three types of memory: texture, main, and disk. As images are acquired by the camera they are loaded into main memory. Feature detection and camera tracking are performed on these images. Images are loaded into texture memory only if requested for texture mapping as part of visualization. Images are written to disk and marked as key-frames only if their associated camera is sufficiently certain and is determined to be novel as above. When storage exceeds a predefined quota for main or texture memory, images are unloaded based on their distance from the current virtual view.

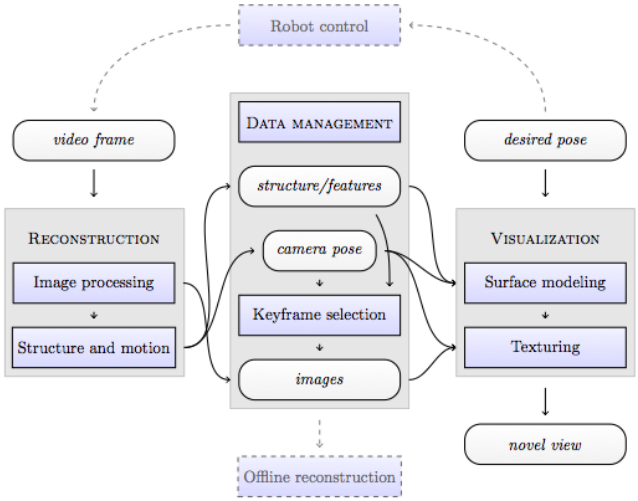


Fig. 6: Predictive Display software system diagram

The visualization thread accesses the most recent geometry and selects keyframes to generate a coarse graphics model at frame rate. At each frame the visualization thread first synchronizes with the reconstruction thread by making a local copy of estimated structure and camera motions/calibrations. Then, to rapidly model a surface for visualization, we select the n key-frames closest to the virtual view, create a view-dependent triangulated surface mesh, and project the key-frame images onto it. See Fig. 9. For selecting the closest cameras we define a distance measure according to [14] that includes the Euclidean distance between camera centres, the angular distance between principal viewing rays, and the distance between the intersections of principle viewing rays and the plane at mean distance to the scene.

To render predicted visual feedback the geometry estimated up to current time is used as a rendering proxy. A texture is computed in one of several ways depending on the situation. Early on in execution the most recent delayed texture image is rendered via the proxy using the current (non-delayed) desired camera view read from the operators master control (here a Phantom). After some time numerous texture frames are available, so the closest frames are blended in graphics hardware. Camera parameters are passed to a shader program which undistorts and projects the raw key-frame images onto the surface geometry.

The texturing of a PD model from multiple keyframes is similar to view-dependent texturing [15]. Here various choices and trade-offs for how to combine the keyframes can be used. A common choice is to texture each model facet (triangle) from the keyframe with the closest matching camera normal (view direction). While texturing each facet from one image gives sharp texture within the facet, there are usually significant artifacts where textures from different keyframes join, and if the geometry is coarse or approximate, artifacts are also at facet boundaries. Other texturing choices involve blending/averaging the pixel colors from several input images. However, this usually introduces blur due to geometry misalignment, see Fig. 7. A more effective way is to modulate a set of basis images, each containing derivatives of the texture images w.r.t. to the warp parameters [6]. The technique of using an image derivative basis to represent small geometric shifts is related to optic flow and SSD tracking, and the tracking and rendering can indeed be integrated in the same framework [16].

V. Experiments

A basic question to answer is how effective is photo-realistic graphical predictive display? To quantify this under controlled conditions we performed user studies on the implemented systems both on the real robot and on a simulated model.

The physical system consists of a WAM robot arm on a Segway mobile base with a camera mounted on the elbow, see Fig. 8. On the operator side, motions are controlled using a Phantom and visual feedback is viewed on a monitor. The operator and remote computer software is linked using PVM (Parallel Virtual Machine). Our WAM does not have a wrist, so for the physical experiments the first 3 DOF of the Phantom are mapped to the robot. For the simulated experiments the full 6 DOF of the Phantom are mapped.

Initial experiments with the visual tracking and model estimation showed that the system could estimate sufficient 3D models for an operator to localize and navigate in the environment. Using the visual tracking the system is robust to disturbances and this was verified by having a person tug the Segway to perturb its position and orientation as illustrated in the video [17]. The visual quality of the rendering from the geometry varies with the change in view direction from texture to view frame. For a moderate change of the type typically needed in PD viewing quality is good. For large

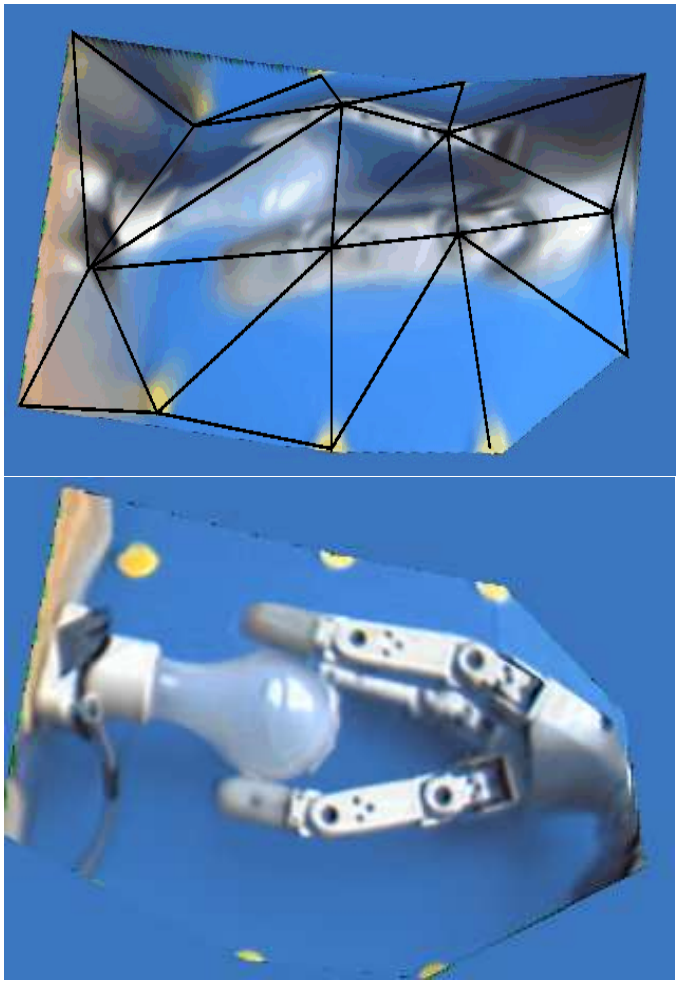


Fig. 7: Top: A rendering from a PD model using color averaging. The model is sparse with just 22 triangles (overlaid) due to the featureless scene. Bottom: Sharper rendering by modulating a texture basis from the same PD model.

changes viewing quality deteriorates, see Fig. 9. Interestingly, while rendering artifacts are obvious when viewing still images, users seem oblivious to these when solving a task using the robotic PD system.

To compare the effectiveness of different predictive display types, several subjects were timed when tele-operating an alignment task. The goal is to match the pose of a target in the scene. Letters were used as targets since they impose a natural order of alignments, A,B,C... and are familiar shapes, however the task is similar to aligning a wrench on a bolt, inserting a module or connector and a variety of other fine manipulation tasks. The task involves a search where the subject has to find where in the environment the target letter is, a reach phase, and a fine alignmnet where the pose of the letter controlled by the user is matched to the one in the scene, see Fig. 10. Four experiment conditions were randomly mixed in the trials: direct video (without delay), 300ms delayed video, simple PD by forward warping the delayed video via a plane, and full PD with a roughly 500 point textured 3D model. The subjects had spent a few

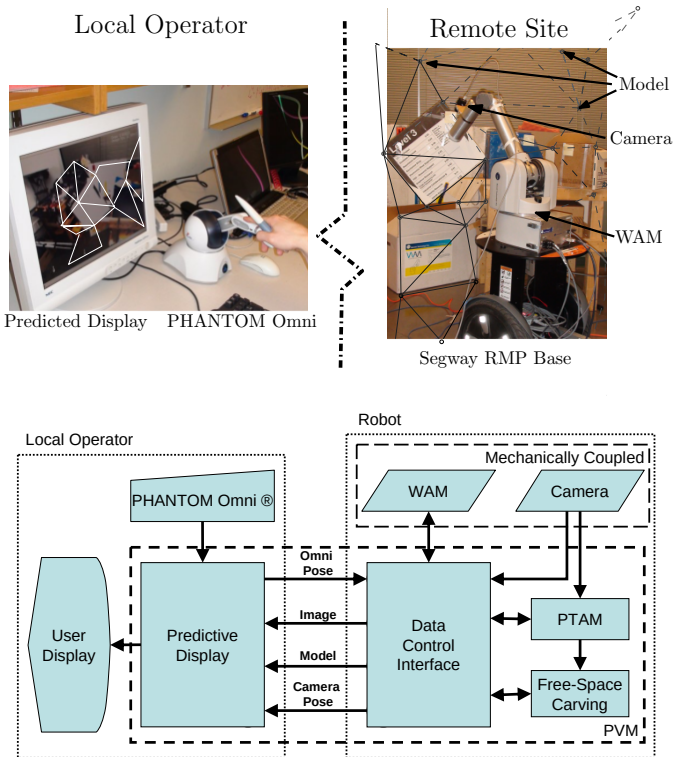


Fig. 8: Top: the operator tele-operates the robot from the local site; the model is computed at the remote site and transferred to the operator for predictive display. Bottom: system components and data flow between the two sites.

minutes before the trials familiarizing themselves with the system.

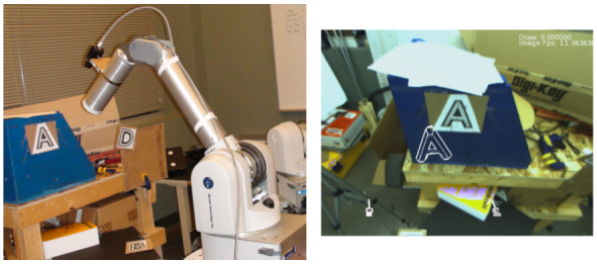


Fig. 10: Align task. Left: image of the robot and camera setup. Right: the operator view showing the overlay (white A) which the operator moves to align with the scene A.

A total of 180 alignments done by 5 users were performed running the physical robot. For experiments using the real robot we had to manually move target objects into various physical test configurations. This is time consuming and because we wanted to measure both mean times and statistical confidence a similar graphics model was created where the alignment target could be automatically positioned.

The more extensive user study based on a simulated environment and had 12 participants performing 24 motion tasks under each of the 4 test conditions for an accumulated total of 1152 trials used to compute the means and confidence

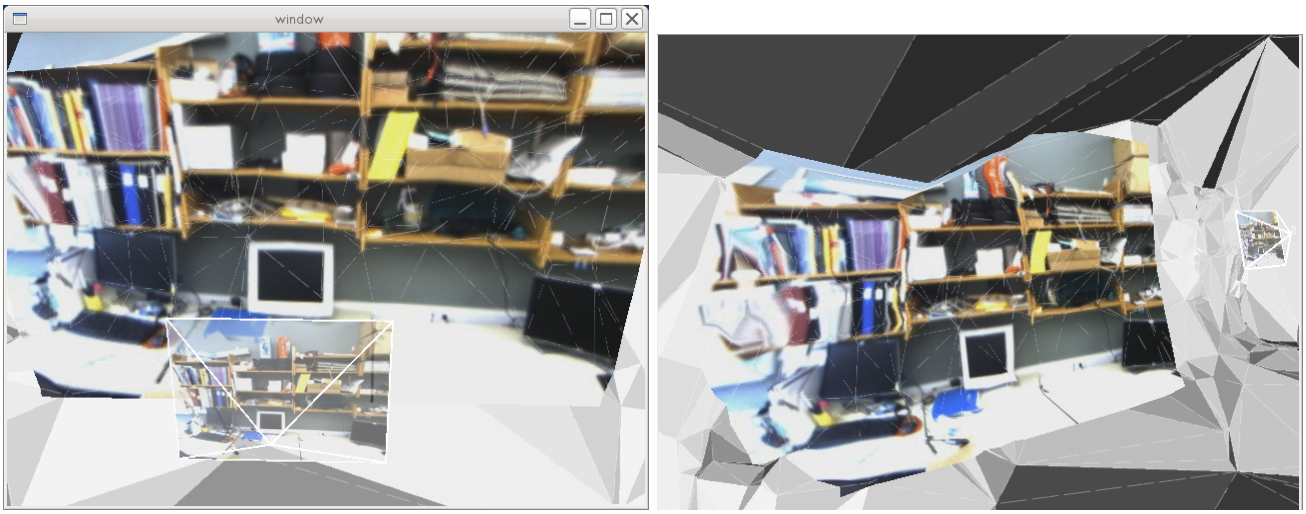


Fig. 9: Left: Texturing from a closeby keyframe. Right: From a significantly different viewangle. Camera location and 3D geometric model illustrated in overlay.

intervals below. For repeatability the same 3D model points were used for all participants, while the view dependent triangulation and texturing were computed on-line. Likewise the texture video frames used were based on the users' exploration of the scene as in a real setting. To be comparable with the real case the speed and angular velocity of the simulated WAM robot was limited to $30cm/s$ and $45°/s$ respectively.

Figure 11 shows the normalized mean time to task completion for each of the letters A,B,C and D. Predictive display always significantly improves time to completion. Aggregating over all trials completing the task with delayed video took 48% longer compared with predictive display. Both PD modes perform overall well and are comparable with the no delay condition. It is worth noting that the performance of the model-based PD improves over time. This is expected as the model improves as more keyframes are added. With more keyframes the user can see a wider field of view and more quickly localize the target.

The graph in Fig. 12 exemplifies the norm of the 6D error residual over a typical trial. The delay case show a move-and-wait strategy (plateaus in the residual) and overshooting (sinusoidal residual) expected in the delayed image mode. Stabilizing plane PD shows some overshooting because the stabilized image is still based on the most recent delayed image. Model-based PD converges relatively smoothly to the target. Some overshooting evident in the no delay residual can be explained by the robot dynamics. The maximum velocity of the robot has some of the same effect as communication delay on task performance. Moving the master to a specific pose is not immediately reflected in the displayed video resulting in overshooting. This also explains why in some specific trials (especially for targets C and D, as explained above), the model-based PD times are better (less time to complete task) than times for no delay.

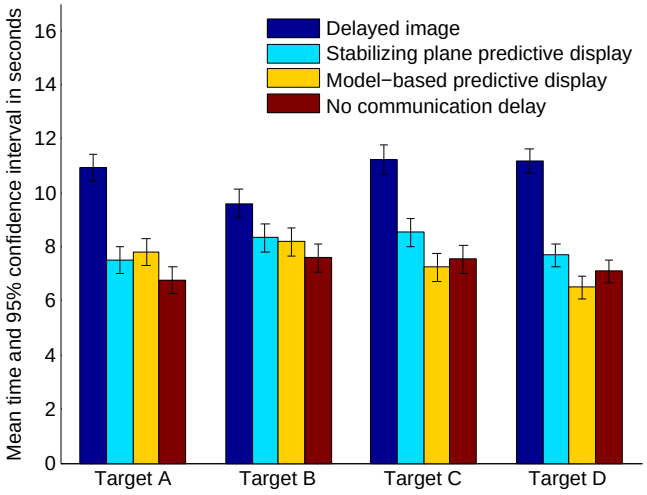


Fig. 11: Time to perform alignment for experts for the first (A) through last (D) alignment target in all six four-target sequences. Within each group, modes with non-overlapping confidence intervals have significantly different mean times.

VI. Discussion and Conclusion

We have designed and implemented a system which acquires a textured graphics model automatically from a robot camera. The system is on-line and unlike most computer vision systems for 3D modeling can provide tele-operator visual feedback immediately using a basic plane warp mode on the delayed video. As the robot manipulation task proceeds more 3D geometry and texture keyframes are acquired it is able to render using a proxy geometry, either by forward projecting delayed video, or in a more advanced way, but composing a view dependent texture from numerous stored keyframes.

Experiments were done with an alignment task, both on

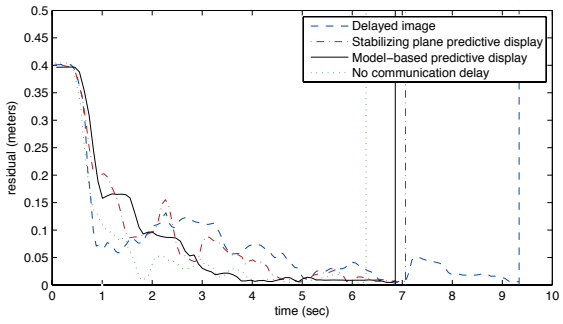


Fig. 12: Error residual for a typical trial

a real tele-robotic system and under controlled conditions where a simulated robot was used. Results over 1200 trials show that even a short 300 ms uncompensated video delay increases task completion time with 48%. Predictive display significantly improved times, and was only 7% off from the no delay case. In the basic task used here, 3D model-based PD had insignificant advantage over plane-transfer PD. In tasks where longer manipulation sequences are involved than the four successive motions used in our experiment it is likely that the advantage of full model based PD over plane based increases since more scene information is incorporated into the model over time.

By comparison to other predictive display systems our task requires a more precise alignment. Similarly our system provides rich lifelike textured and graphics rendered predictive display, while older systems used wireframe overlay drawings from a-priori CAD models. It is likely that with increasing complexity of tele-robotics tasks, better predictive display is needed. What delays are tolerable for a human operator depends on the task, but is in general quite low, below 0.3s for hand-eye coordination and even less for viewpoint/head reorientations when e.g. wearing a head mounted display. Overall, to provide a sense of presence and fidelity in tele-manipulation, both the actuation system (robotic master and slave) and the sensory feedback system (video, predictive display and possibly haptics) matter.

REFERENCES

[1] P. Hokayem and M. Spong, "Bilateral teleoperation: An historical survey," in *Automatica*, vol. 49, no. 12, December 2006.

[2] R. Held, A. Efstathiou, and M. Greene, "Adaptation to displaced and delayed visual feedback from the hand." *J. Exp Psych*, vol. 72, pp. 871–891, 1966.

[3] T. Sheridan, "Space teleoperation through time delay: review and prognosis," *Robotics and Automation*, vol. 9, no. 5, pp. 592–606, 1993.

[4] A. Bejczy, W. Kim, and S. Venema, "The phantom robot: predictive displays for teleoperation with timedelay," in *Robotics and Automation, 1990*, vol. 1, May 1990, pp. 546–551.

[5] T. Kotoku, "A predictive display with force feedback and its application to remote manipulation system with transmission time delay," in *Intelligent Robots and Systems, 1992*, vol. 1, July 1992, pp. 239–246.

[6] K. Yerex, D. Cobzas, and M. Jagers, "Predictive display models for tele-manipulation from uncalibrated camera-capture of scene geometry and appearance," in *in IEEE International Conference on Robotics and Automation(ICRA*, 2003.

[7] T. Burkert, J. Leupold, and G. Passig, "A photorealistic predictive display," *Presence: Teleoperators and Virtual Environments*, vol. 13, no. 1, pp. 22–43, 2004.

[8] A. Rachmielowski, N. B. D. Cobzas, and M. Jagersand, "Performance evaluation of monocular predictive display," in *Proc. of IEEE ICRA*, 2010.

[9] M. Vergauwen and L. V. Gool, "Web-based 3d reconstruction service," *Mach. Vision Appl.*, no. 17, pp. 411–426, 2006.

[10] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, 2007.

[11] A. Rachmielowski, D. Cobzas, and M. Jagersand, "Robust SSD tracking with incremental 3D structure estimation," in *Canadian Conference on Computer and Robot Vision*, 2006, pp. 1–8.

[12] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

[13] D. Lovi, N. Birkbeck, D. Cobzas, and M. Jagersand, "Incremental free-space carving for real-time 3d reconstruction," in *Proc. of 3DPVT*, 2010.

[14] J. f. Evers-senne and R. Koch, "Image based interactive rendering with view dependent geometry," in *Eurographics*, 2003.

[15] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," *j-COMP-GRAPHICS*, vol. 30, no. Annual Conference Series, pp. 11–20, 1996. [Online]. Available: http://www.acm.org:80/pubs/citations/proceedings/graph/237170/p11-debevec/

[16] D. Cobzas and M. Jagersand, "Tracking and predictive display for a remote operated robot using uncalibrated video," in *Proc. of IEEE ICRA*, 2005, pp. 1847–1852.

[17] "Movies of the experiments are available on http://www.cs.ualberta.ca/∼vis/pd."