A Panoramic Model for Remote Robot Environment Mapping and Predictive Display

Dana Cobzas, Martin Jägersand and Hong Zhang
Computing Science, University of Alberta, Canada
{dana, jag, zhang}@cs.ualberta.ca

Abstract: Robot tele-operation is significantly degraded by delays in the operator visual feedback. We present a cylindrical image-depth model that can be automatically acquired at the remote work site and then used to support both robot localization and predictive display. We present an implementation for a mobile robotics system where synthesized immediate visual feedback replaces the delayed real images. Experimentally we found that a predicted view could be rendered with an geometric viewpoint error of no more than 2.4cm/1.2 degrees in a 100m^2 work scene.
Keywords: Image-based modeling, Robot navigation, Panoramic model, Predictive display.

1 Introduction

An important and challenging problem in tele-robotics is how to convey the situation at a distant robot site to a human operator. Efficient and accurate tele-operation makes use of two types of feedback. Real-time high-fidelity visual feedback qualitatively conveys the remote situation in an intuitive way. A metric model of the remote scene combined with a real-time localization method allows the operator to make decisions and base motions on quantitative measurements.

In real-world applications such as emergency response, unmanned exploration, and contamination cleanup, typically the remote scene geometry is unknown, and the data link between the robot and operator can have both delays and limitations in capacity. Hence, at the outset, the operator has not only little information about the remote scene, but also limited means of transmitting timely high fidelity video and other data. Yet, research has shown that tele-operation performance is significantly degraded with delays of as small as 0.4 seconds [10]. This causes operators to lose direct intuitive connection between their control actions and effects at the remote site displayed in the video feedback.

Both quantitative and qualitative information from a remote tele-robotics scene can be improved through the use of geometric modeling. For quantitative robot localization, typically a 2 or 3D Cartesian world model is used. Unfortunately, without *a-priori* knowledge about the environment, it is in general difficult to create a precise metric map at a level of detail sufficient for robot localization [16, 1]. More recent vision-based localization systems use geometric features, obtained from a variety of sensors (ultrasound, laser, vision) and integrated into a probabilistic map [23, 7]. Image patches that have a distinct signature can be matched from one frame to the next using either PCA encoded [19] or SIFT landmarks [17].

A different approach to geometric feature-based maps are the *appearance maps* that are created by "memorizing" the navigation environment using images or templates. By comparing the templates in the model with its current view, a robot can derive control commands to steer itself along a memorized route [13] or to a goal position [11, 14, 24]. One of the major drawbacks of these appearance-based maps is that robot motion is restricted to either a predefined route or positions close to the locations where the images were originally acquired.

By contrast, in image-based computer graphics the objective is to generate models which represent the visual appearance of a scene from all view points. In model-based approaches a detailed geometric scene model, and several photos registered with the scene are used to render new views[20]. Rendering based on a lumigraph on the other hand represents the ray set (plenoptic function) on some surface, and hence does not need exact knowledge of the scene geometry[9, 12], but instead uses exactly knowledge of the camera position to determine the ray set. By representing the plenoptic function on a cylinder (panoramic image) views can be synthesized for different directions, but only from the same viewpoint[21].

In mobile robotics obviously different viewpoints need to be represented. However, current localization algorithms do not give precise enough pose to integrate the views from several positions into one lumigraph. We present an approach in-between image and geometry based modeling. A cylindrical panoramic image is accurately captured by rotating a camera about its optical center. The image information is augmented by depth values obtained from

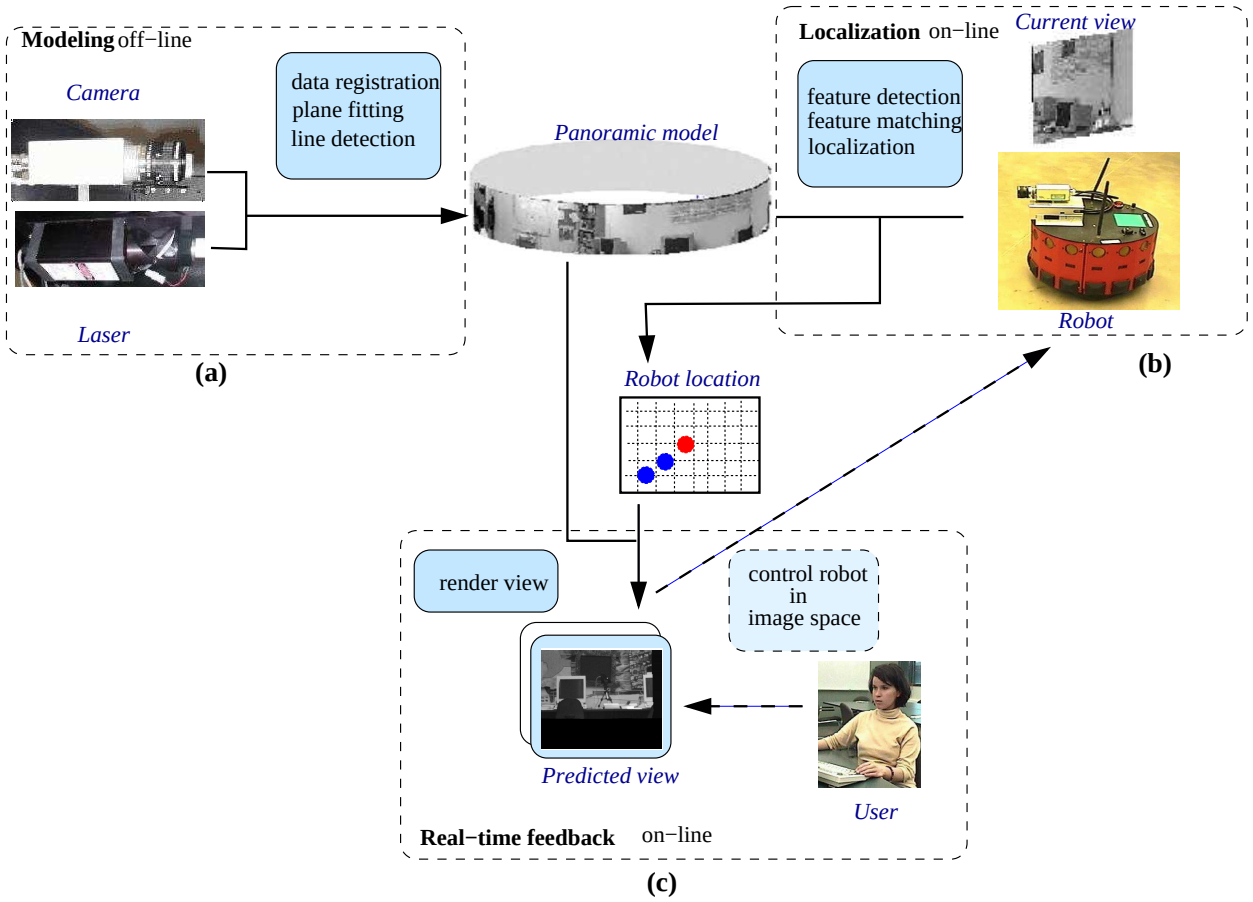


Figure 1: Overview of the navigation system with predictive display. The image-based panoramic model is acquired and processed off-line using a camera and a laser range finder (a). The on-line localization system matches lines features in the model with the ones extracted from the current view (b). The robot location is send to the remote site where a predictive view is immediately rendered (c). The remote user can control the robot using the image-based interface.

a laser range finder attached to the rotating camera. As we have shown in [6], this model supports robot localization. In this paper we present a method to refine the cylindrical model into a piece-wise planar format suitable for rendering, and then implement a predictive display system for tele-robotics.

In predictive display two qualities of the model are essential: The ability to determine the correct view point (location) to render, and the support of high fidelity image rendering of that view point. Most current systems use geometric CAD models to support rendering and calibrated additional sensors (e.g magnetic or joint angle) to determine pose [18]. However, this is impractical in unstructured environments often encountered in mobile robotics. A better method to align the predicted display is to register the remote camera image with the model [2]. In our approach the model is obtained once through automatic sensing as described above, then a single camera image is used to align the current view location with the model to support both localization and predictive display. An overview of our system is presented in Figure 1. Specifically, the model is formed by a panoramic image-based model augmented with a sparse set of 3D vertical line features (a) as described in Section 2. This model contains sufficient information about the navigation environment without explicit full 3D reconstruction. The model acquisition and processing is performed off-line once. When the room map has been obtained, we use an incremental on-line localization algorithm that matches the line features in the robot's current camera view, with the line features in the navigation map to estimate the robot's position (b) (this is described in [6]). After robot position has been obtained, it is sent to the remote location to generate a synthesized view using the same model (c) as described in Section 3. In this way, the operator that

is controlling the robot has a user friendly interface as shown in the experiments, Section 4.

2 Image and Depth Model

Our image-based model consists of a panoramic image mosaic that is registered with range data acquired by a laser range finder (see Figure 2). The main steps in building the model are:

```
(1) Data acquisition
      panoramic mosaic acquisition
         (rotating camera)
      range data acquisition
         (rotating laser rangefinder)
(2) Data registration
      range data filtering
      range data cylindrical representation
      global registration (image to image warp)
(3) Robust plane fitting
      edge detection and linking
      constraint Delaunay triangulation
      region growing based on fitted plane
      data projection on fitted planes
(4) Vertical line detection
      vertical segments in panoramic mosaic
      3D line parameters
```

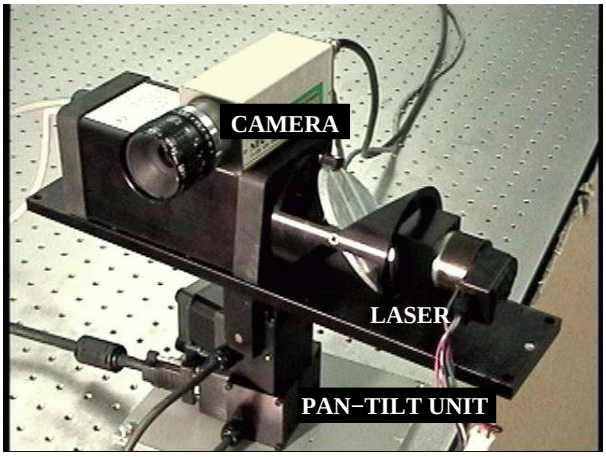


Figure 2: System configuration: the laser-rangefinder with the camera attached on top of it is mounted on a pan-tilt unit

Step 1: Data acquisition

The model is acquired using a *laser rangefinder* (Acuity Research, Inc.) and a CCD *camera*, mounted on a *pan-tilt unit (PTU)* (see Figure 2). We use the pan axis of the PTU to rotate the camera in order to build a cylindrical or panoramic image model. A 180° panoramic mosaic is shown in Figure 3. The pan axis of the PTU and a rotating mirror attached to the laser range finder produce two degrees of freedom, spanning a unit sphere. The range data is registered with respect to the center of rotation and represented as a spherical image.

Step 2: Data registration

After acquiring the panoramic image mosaic and the spherical range data set, they need to be registered with respect to a common reference frame. The registration of volumetric and intensity data is an important problem especially in the fields of model building and realistic rendering. Most of the proposed solutions recover the rigid transformation between the sensors using point or line features (e.g. [20]). This is in general a non-linear problem and requires a good initial estimate to converge. In contrast, image-based techniques compute a direct mapping between the points in the data sets to recover the transformation. In a recent paper [5] we compared the two approaches and found that image-based methods are fast and adequate for application that does not require a high precision alignment. In the setup presented here, the two sensors are very close to each other, and an image to image warp is suitable for aligning the two data sets. The spherical range image is converted to a cylindrical representation with the radius equal with the focal length of the camera. Then a *image mapping* is established between the intensity and range data in similar cylindrical image representations. The detailed registration algorithm is described in [6].

Step 3: Robust plane fitting

Man-made environments are mostly composed from planar regions. A common technique in data modeling [20, 22] is to segment the range (3D data) into planar regions. In most of the cases, the goal is to create a 3D CAD-like model of the scene, composed of planar regions that are suitable for rendering. In our case, we do not reconstruct a full 3D model of the scene, but extract a sparse set of robust 3D features (e.g. lines) that are required by the localization algorithm. So we use the plane fitting algorithm to eliminate bad data and create a cleaner model.

For planar region detection, we used an algorithm inspired by [8, 4]. It starts with a set of triangles in image space and merges these into larger regions based on residual error from a best fitted plane to the corresponding 3D data points.

In a typical indoor environment, planar regions are often bordered by intensity discontinuities (edges in the image). Based on this observation, our algorithm starts with a 2D mesh of triangles generated in the panoramic mosaic by a constrained Delaunay triangulation with edge segments. From the initial triangular regions, a *region adjacency graph* is created, where the vertices represent the regions and the edges indicate that two regions are adjacent. Each edge is weighted by the residual error of a plane robustly fitted to the union of the 3D points corresponding to the adjacent regions R_i,R_j that share that edge.

$$E_{i,j} = \sum_{\mathbf{P}_k \in R_i, R_j} \alpha_k (\mathbf{n}^T \mathbf{P}_k + d)^2 \qquad (1)$$

where $\mathbf{P}_k$ are corresponding 3D laser points for the

Figure 3: Spherical representation of the range data from an 180° scan after filtering (top). Corresponding 180° panoramic image mosaic (bottom).

regions and α_k is a weight factor (see the paragraph about plane fitting). Larger regions are grown from the initial mesh by merging, at every step, adjacent regions that produce the smallest error. This guarantees that the total error grows as slowly as possible. We use a threshold on the total number of regions as the stopping criterion. At the end we project the 3D points in each region to the corresponding fitted plane. Figure 4 illustrates the plane detection algorithm on a segment of the panoramic image: original image (a), detected edge segments (b), mesh triangles (c), and final planar regions (d).

The plane equation is $\mathbf{n}^T\mathbf{P} + d = 0$, where $\mathbf{n}$ is the plane normal, and d represents the distance from the origin to the plane. With N points, a least square plane fitted to the points can be estimated by minimizing the error

$$\min \sum_{k=1}^{N} (\mathbf{n}^T\mathbf{P}_k + d)^2 \tag{2}$$

By normalizing the points with respect to their centroid $\mathbf{P} = \frac{1}{N}\sum_{k=1}^{N}\mathbf{P}_k$, we obtain the zero mean points $\mathbf{A}_k = \mathbf{P}_k - \mathbf{P}$, and the problem can be rewritten as

$$\min \sum_{k=1}^{N} \mathbf{n}^T\mathbf{A}_k \tag{3}$$

The solution to this problem $\mathbf{n}_{min}$ is an eigenvector of length one of the covariance matrix $\mathbf{\Lambda}$ associated with the smallest eigenvalue. The covariance matrix has the form

$$\mathbf{\Lambda} = \frac{1}{N}\sum_{k=1}^{N} \mathbf{A}_k\mathbf{A}_k^T$$

The distance to the best fitted plane is given by

$$d_{min} = -\frac{1}{N}\sum_{k=1}^{N} \mathbf{n}_{min}^T\mathbf{P}_k \tag{4}$$

It is well known that even a few outliers present in the data set can cause an erroneous estimation of the fitted plane. We adopted a robust fitting algorithm that first estimates a plane as described before and then assigns weights to the points depending on their residual:

$$\alpha_k = 1/(\mathbf{n}^T\mathbf{P}_k + d)^2$$

The plane is re-estimated by solving the weighted least square problem

$$\min \sum_{k=1}^{N} \alpha_k(\mathbf{n}^T\mathbf{P}_k + d)^2 \tag{5}$$

In practice we eliminate points that have residuals above some threshold ($\alpha = 0$) and re-estimate the plane with the remaining ones ($\alpha = 1$).

Step 4: Vertical line detection

The robot localization algorithm matches 3D vertical line features from the model with detected vertical edges in the current image. To calculate the model lines' parameters, we first detect vertical edges in the panoramic model and then the corresponding 3D points from the registered range data. There are three major categories of discontinuities in 3D that can generate an edge in the image: (a) lines on the same surface that are discontinuities in intensity (color), (b) surface discontinuities, and (c) occluding contours on a smooth surface. In our algorithm we used only the first case where the line points belong to the same planar region, and a unique line can be calculated by fitting the line equation to the 3D points.

Finally, to produce the composite model, the image-based navigation map that is used for robot navigation and predictive display consists of the panoramic image mosaic registered with a sparse, piece-wise planar 3D model, and a set of vertical line segments with corresponding 3D coordinates.

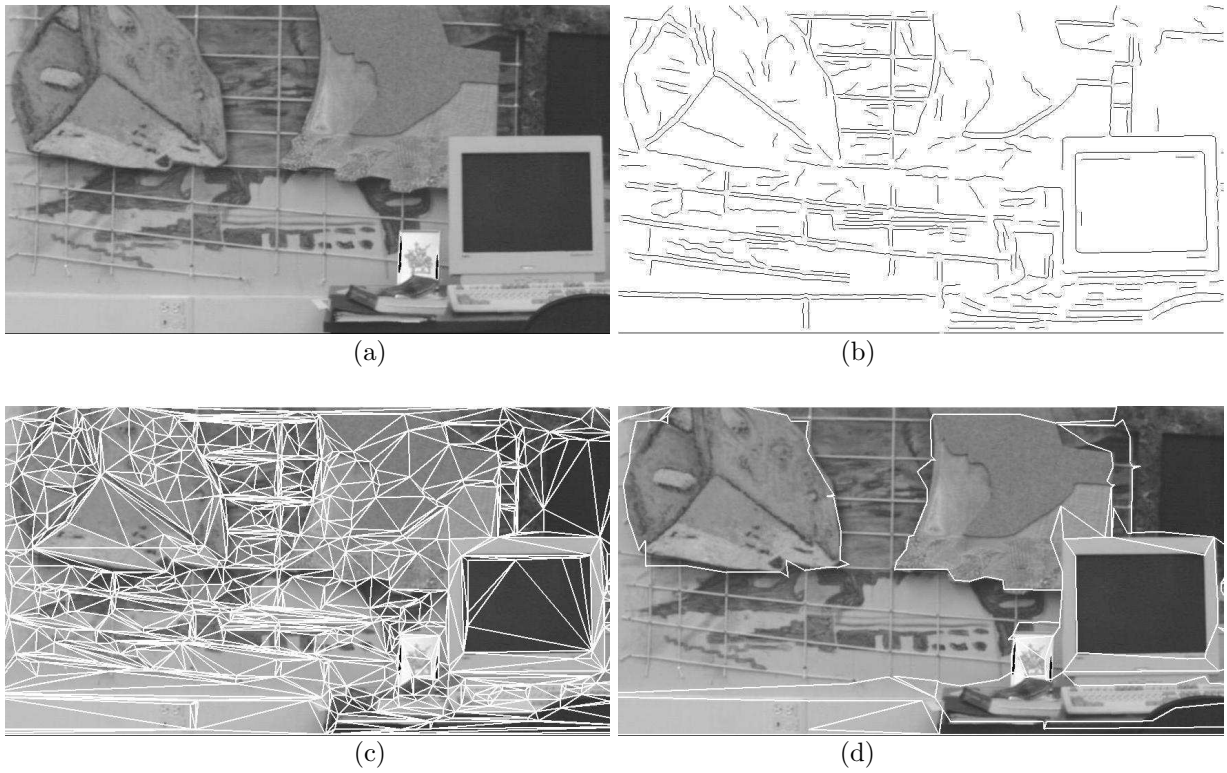


Figure 4: Planar region segmentation: (a) Original image; (b) Edge detection and linking; (c) Constraint Delaunay triangulation; (d) Merged planar regions

3 Predicted Scene Rendering

In tele-robotics a robot at a remote site is controlled by a human operator. Good operator and system performance depends on the tele-robotics system's ability to visually immerse the operator in the remote scene. In particular, it is necessary to ensure that round trip delay between an operator motion command and the visual feedback received by the operator from the remote site must be minimized. It has been shown that longer delays prevent complex motion tasks, and operators adopt inefficient "move and wait" strategies to motion control [3]. While time delays are inherent in information transmission over a distance, the effect of the delays can be reduced by synthesizing estimated images based on our model, and showing this video for operator visual feedback. In the current implementation, we used the model to synthesize the current robot view from the acquired model ahead of the arrival of actual (delayed) scene video. Three components are needed to render a predicted view:

1. A geometric model of the scene segment to be rendered. We use the piece-wise planar model derived in Section 2.
2. A viewpoint for the predicted image. From our model we can derive the robot location as shown in [6], (showing current motion) and to it add the operator motion command (predicting the outcome of the next action ahead of its completion).
3. A texture image to warp onto the geometric model. The predicted view can be textured either from the panorama or a previous camera image.

The image-based panoramic model with registered sparse range values are stored at both robot and tele-operator locations. We used as a geometric model of the scene the triangulation mesh formed by a constrained Delaunay triangulation with edge segments in the panoramic mosaic. This triangular mesh was the starting point of the region growing algorithm in Section 2 (see Figure 4 (c)). For each vertex in the panoramic image $\mathbf{p}_c$, we compute its corresponding 3D coordinate $\mathbf{P}$ (in model coordinate system) by interpolating the existing 3D laser points in its vicinity.

We developed an OpenGL hardware accelerated implementation that allows rendering in real-time. Given a viewpoint, the geometric model is projected in the virtual camera and textured with an image from the remote scene. The panoramic image contains the textures for all viewing directions, but from

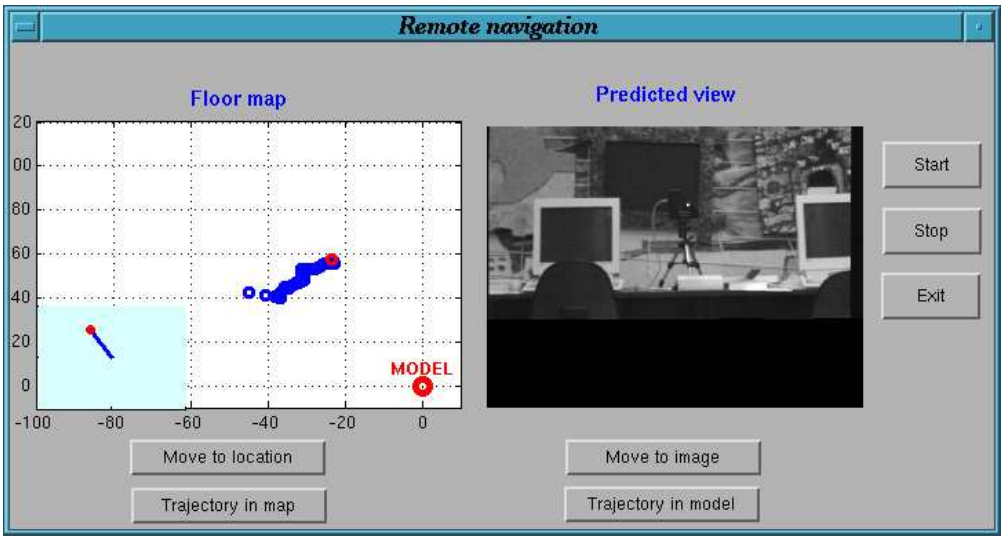


Figure 5: Tele-operator interface where the user can view a 2D floor map with robot position and the predictive robot view. The vector in the small left window shows robot orientation.

a single viewpoint. This provides a good approximation to textures also in nearby viewpoints. For rendering viewpoints far away from the model center the panoramic texture has several drawbacks (e.g. significant quantization effects due to aliasing, occlusions, amplification of errors in the geometric model). A better alternative is to instead use the last received image from the robot site. Through the localization procedure it is registered with the model and then re-warped into a new view. Since the motion of the real robot is continuous and smooth most of the new view can usually be textured from a previous delayed image. Only during large rotational motions where the robot rotates with an angular speed approaching the transmission delay does the limited viewing angle of the real camera cause significant portions of the predicted view to not be textured. The process of generating the predictive view can be summarized as follows.

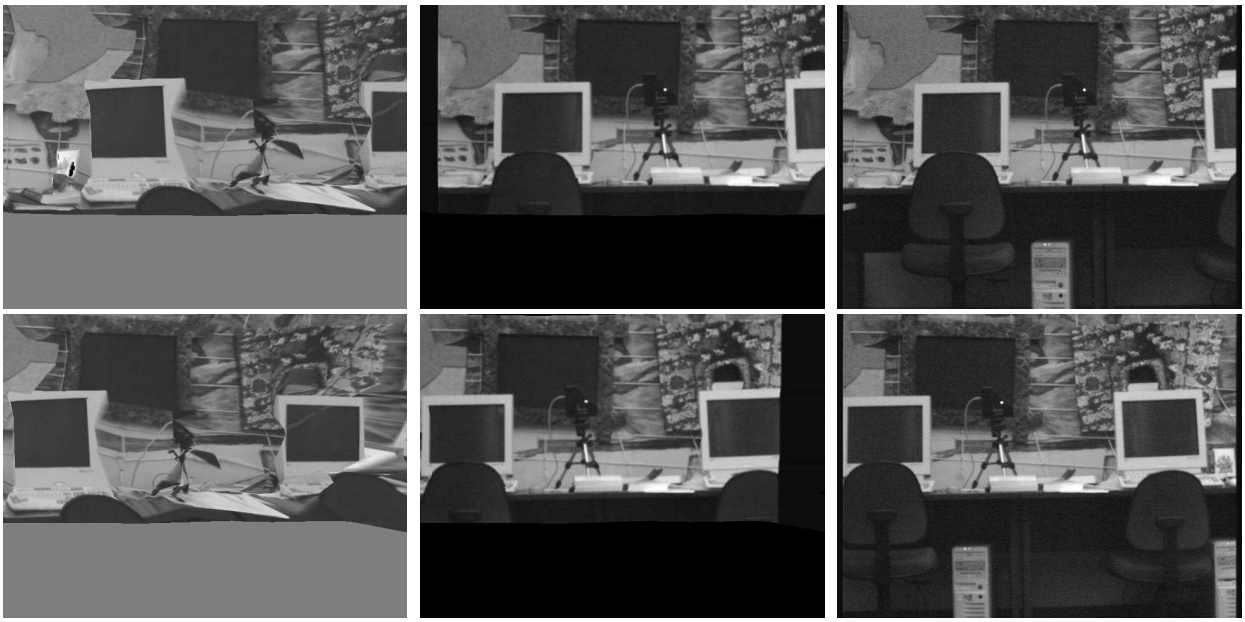
```
for each time step t
    robot site:
      (1) calculate robot location (R_y(t), t(t))
      (2) send position to operator site
      (3) add current operator motion command
      (Asynchronously) Send robot camera image
    user remote site:
      (4) project scene model points P
            p(t) = C(R_y(t)P + t(t))
      (5) generate synthesized robot view with
            (a) texture from panoramic model image
            (b) or texture using delayed
                image from t - 1
end for
```

In **Step 1**, the robot location (position and orientation) is calculated from the robot view in terms of a 2D image. We assume planar motion, which is reasonable for indoor environments where motion takes place on the floor. We developed an on-line incremental localization algorithm [6] that is matching the model lines with detected vertical edge segments in the current robot view. From corresponding pairs of line segments we compute robot orientation (pan angle) $R_y(t)$ and position on the floor plane (2D translation) $\mathbf{t}(t)$.

After the robot position has been calculated, it is sent to the remote site in **Step 2**, the current incremental operator motion comand between $t-1$ and t is added in **Step 3**, and a predictive view can be generated by projecting the geometric model in the new location in **Step 4**.

In **Step 5**, texture mapping a geometric model assumes that it is decomposed in triangular planar surfaces. In a complete mesh, some of the triangles might not represent physical planes but are artifacts of occluding contours. In most cases, these appear as silhouette edges [15] where points from the object are connected with background points. To avoid this phenomenon, we eliminate all the triangles that (within a threshold) are parallel to the viewing direction.

Texturing the predicted view from the cylindrical panorama can be done directly, since the registration provides the texture coordinates for each model vertex. In the cases where a previous camera image $I(t-1)$ from a different view point is used the correct texture coordinates are obtained by projecting the geometric model using the robot location at time $t-1$, $(R_y(t-1), \mathbf{t}(t-1))$. In this way the delayed image $I(t-1)$ is warped into the new view location to generate a predictive view at time t. Assuming



(a) texture from panorama (b) texture from previous image (c) ground truth (delayed image)

Figure 6: Examples of predictive views using different texturing

a continuous and smooth motion, we can generate real-time realistic views along the entire robot trajectory.

4 Experimental Results

We integrated the localization algorithm and the predicted display into a common interface where the remote user can visualize both robot position on a 2D floor map and a see a predictive synthesized robot view (see Figure 5). In addition to forward predicting and rendering views one or a few time steps ahead of the robot, using the model and panoramic image, the operator can also visualize entire simulated robot motion trajectories in the remote scene before executing the actual robot motion. The navigation system can be extended by offering the user the possibility to control the robot by dragging and clicking on desired locations in the rendered view. This will offer a natural and intuitive way to control the robot without having to interpret the geometric map.

To evaluate the model accuracy and performance of the localization algorithm with predictive display, we acquired a panoramic image and range model from the Center for Intelligent Mining (CIMS) lab (see Figures 3 and 4). Comparing to ground truth (obtained by measuring and marking locations on a floor grid in the $100m^2$ lab) we found that localization and predictive view rendering could be performed with an average 2.2 cm translational and 1.2 degrees rotational accuracy of the view points. In evaluating the on-line forward prediction from delayed images, 26 robot camera images along a trajectory were forward warped to viewpoint about 10cm ahead on the trajectory and compared to a real image obtained at that same viewpoint. Figure 6 shows examples of rendered views along robot trajectory using texture from panoramic model (middle column) and previous (delayed) real image (left column). Notice that as mentioned in Section 3, in texturing from the cylindrical panoramic image, any pan angle can be rendered, while when texturing from a delayed image, only the overlap in pan angle between the two views can be rendered. This accounts for the black stripe to the left or right of the predicted views in the middle column. The black stripe in the bottom is due to the cylindrical model being cut off at that level. Comparing the predicted with real views (in the right column), we notice that the rendering from delayed images produces better quality visual feedback than rendering from the panorama. This is to be expected because between two successive robot views there is only a small displacement, so distortions caused by texture or geometry errors are minor.

5 Conclusions

A main consideration in designing robotic teleoperation systems is the quality of sensory feedback provided to the human operator. For effective teleoperation the operator must get the feeling of being present in the remote site and get immediate visual feedback from his or her motion commands. While in consumer applications of image rendering, the most

important criterion may be the subjective pleasantness of the view, for accurate robot control the geometric precision of the rendered viewpoint is more important than minor errors in scene surfaces or textures. Our system synthesizes the current view of the robot using only position information, which is calculated with a centimeter accuracy. In a low bandwidth system, this avoids the delay introduced by sending the full robot image. By also adding the current operator motion command to the pose estimate, local predictive display is synthesized immediately in response to operator command. This provides direct visual feedback to the operator movement, avoiding both the latency and bandwidth introduced delays in remote scene communication.

Our model directly relates geometric robot pose and image views, and this also can support control interfaces where the motion goal is specified in image space instead of robot motor space. One such possible intuitive interaction paradigm is tele-operating the robot by "pointing" in the image space or by dragging the model viewpoint to obtain the desired next view, and then have the robot move to this location using visual servo control.

References

[1] N. Ayache and O. D. Faugeras. Maintaining representation of the environment of a mobile robot. *IEEE Transactions on Robotics and Automation*, 5(6):804–819, 1989.

[2] M. Barth, T. Burkert, C. Eberst, N.O. Stöffler, and G. Färber. Photo-realistic scene prediction of partially unknown environments for the compensation of time delays in presence applications. In *Int. Conf. on Robotics and Automation*, 2000.

[3] Antal K. Bejczy, Won S. Kim, and Steven C. Venema. The phantom robot: predictive displays for teleoperation with time delay. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 546–551, 1990.

[4] D. Cobzas and H. Zhang. Planar patch extraction with noisy depth data. In *Proc. of 3DIM*, 2001.

[5] D. Cobzas, H. Zhang, and M. Jagersand. A comparative analysis of geometric and image-based volumetric and intensity data registration algorithms. In *Proc. of IEEE ICRA*, pages 2506–2511, 2002.

[6] D. Cobzas, H. Zhang, and M. Jagersand. Image-based localization with depth-enhanced image map. In *Summited to IEEE ICRA*, 2003.

[7] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.

[8] O. Faugeras, B. Hotz, H. Mathieu, P. Fua Z. Zhang, E. Theron, L. Moll, G. Berry, J. Vuillemin, P. Bertin, and C. Proy. *Real time correlation-based stereo algorithm, implementations and applications.* INRIA Technical Report No. 2013, 1993.

[9] S. J. Gortler, R. Grzeszczuk, and R. Szeliski. The lumigraph. In *Computer Graphics (SIGGRAPH'96)*, pages 43–54, 1996.

[10] R. Held, A. Efstathiou, and M. Greene. Adaptation to displaced and delayed visual feedback from the hand.

[11] J. Hong, X. Tan, B. Pinette, R. Weiss, and E. Rseman. Image-based homing. *IEEE Control Systems*, pages 38–44, 1992.

[12] M. Levoy and P. Hanrahan. Light field rendering. In *Computer Graphics (SIGGRAPH'96)*, pages 31–42, 1996.

[13] S. Li and S. Tsuji. Qualitative representation of scenes along route. *Image and Vision Computing*, 17:685–700, 1999.

[14] Y. Matsumoto, M. Inaba, and H. Inoue. Visual navigation using view-sequenced route representation. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 83–88, 1996.

[15] D. K. McAllister, L. Nyland, V. Popescu, A. Lastra, and C. McCue. Real-time rendering of real world environments. In *Proc. of Eurographics Workshop on Rendering*, Spain, June 1999.

[16] H. P. Moravec. The stanford cart and the CMU rover. *Autonomous Robot Vehicles*, pages 407–419, 1990.

[17] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2051–2058, 2001.

[18] T. B. Sheridan. Space teleoperation through time delay: Review and prognisis. *IEEE Tr. Robotics and Automation*, 9, 1993.

[19] Robert Sim and Gregory Dudek. Learning and evaluating visual features for pose estimation. In *ICCV*, pages 1217–1222, 1999.

[20] I. Stamos and P. K. Allen. 3D model construction using range and image data. In *Proc. of CVPR*, 2000.

[21] R. Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, pages 22–30, March 1996.

[22] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment map s. In *Computer Graphics (SIGGRAPH'97)*, pages 251–258, 1997.

[23] Sebastian Thrun, Maren Bennewitz, Wolfram Burgard, Armin B. Cremers, Frank Dellaert, Dieter Fox, Dirk Hahnel, Charles R. Rosenberg, Nicholas Roy, Jamieson Schulte, and Dirk Schulz. Minerva: A tour-guide robot that learns. In *KI - Kunstliche Intelligenz*, pages 14–26, 1999.

[24] N. Winters and J. Santo-Victor. Mobile robot navigation using omnidirectional vision. In *Proc. 3rd Irish Machine Vision and Image Processing Conference (IMVIP'99)*, 1999.