Catadioptric Image-based Rendering for Mobile Robot Localization

Hynek Bakstein
Center for Machine Perception, FEE
Czech Technical University in Prague
Karlovo nám. 13, Prague, Czech Republic
bakstein@cmp.felk.cvut.cz

Aleš Leonardis
Visual Cognitive Systems Laboratory
Faculty of Computer and Information Science
University of Ljubljana, Slovenia
http://vicos.fri.uni-lj.si

Abstract

We present an approach to view-based mobile robot localization using a X-slits image based rendering (IBR) method for creating novel views from a set of input images. The input images are acquired by a non-central catadioptric sensor mounted on a robot moving on a straight line. We propose to use the IBR for column ordering only, where occlusions in the horizontal direction are modeled and the sensor can be non-central. For the column matching between a query view at an unknown position and virtual views created by IBR, we use correlation of columns.

1. Introduction

Robot localization is an important task when the robot has to find its position in a known environment using some representation of the environment. There are two localization scenarios; the 'kidnaped' robot problem and incremental localization. The 'kidnaped' robot problem is formulated as reentering a previously visited environment at an unknown location and the robot has to localize without any prior hint of its position. In incremental localization, the robot looks for an unknown position is some small neighborhood area of its previous location.

We focus on the case of purely visual representation and the kidnaped robot scenario, when the robot uses a query view of the environment captured at unknown location and tries to extract its position from images describing the environment. We also restrain ourselves from using any markers deployed in the scene. For this scenario, there are basically two approaches; feature and image based methods. Feature based approaches [4, 7, 8, 10] use natural landmarks in the environment, perform matching between landmarks detected in the query view with landmarks in the visual representation, and triangulate the position of the robot. Image based methods [6, 3] use matching of the whole query view with images captured at known positions in the scene — the visual representation of the environment. The closest view gives the position of the robot.

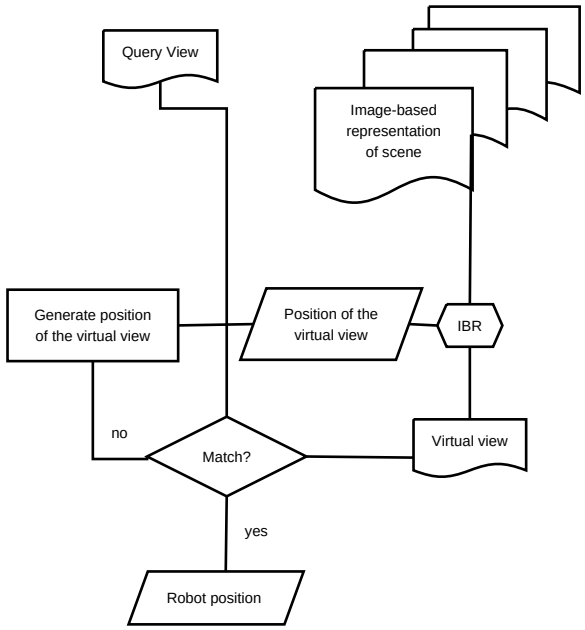

Figure 1. Robot localization using IBR.

Feature based approaches need enough stable, unambiguous, features to be detected in the scene and they also require a central camera model for triangulation (or calibrated non-central camera) while image based methods are very sensitive to occlusions. Our proposed image based rendering (IBR) approach to localization tries to combine the strength of image based methods with robustness to occlusions. IBR [9] creates novel images from a set of input images by recomposition of pixels into the novel view. The novel view appears as captured from a camera position not contained in the input set. We pose the robot localization as an optimization problem, where we look for *the most similar virtual view* to some query view when the parameters used to synthesize the virtual view determine the position of the robot. The localization process is depicted in Figure 1.

We have chosen X-slits method for IBR [12] because it uses only a very simple constant scene depth assumption and the position of the virtual camera can be directly related to the location position of the robot with respect to the input set.

The most related work to the presented method is [11], where the localization was performed from 1D panoramic images representing the horizon line in catadioptric images. Our approach differs in employment of much more information in the matching for robustness and it also does not require the horizon to be precisely determined. In [1], X-slits IBR was used for mobile robot localization with a narrow-field-of-view camera, which suffers from aperture problem. Moreover, employment of a camera with narrow FOV results in a much simpler implementation of X-slits IBR but only a single viewing direction can be used for all input images as well as for the query images. Omnidirectional images capture light rays in all directional and the rotation between the query view and the input views can be estimated [5].

The paper is organized as follows. In Section 2 we summarize omnidirectional X-slits rendering with catadioptric images. Section 3 presents the idea of IBR as a column ordering task, with the column matching presented in Section 4. Experimental evaluation is given in Section 5.

2. Catadioptric X-slits images

X-slits [12] is an imaging model where all light rays forming the image intersect two, generally distinct, lines (slits). One of these lines is the trajectory of the camera which captured the input images, the other corresponds to the position of the virtual view. It is convenient to ensure that the lines are perpendicular in the X-slits setup, which leads to a simplified image formation model, where novel images are created by recomposition of columns of the input images [12]. X-slits model for panoramic images was also introduced in [12] but it was not discussed in detail. In this section, we give a thorough analysis of panoramic X-slits with emphasis on localization tasks.

Figure 2 depicts the principle of panoramic X-slits. Input panoramic images are acquired on a linear trajectory and a novel view I_v is created by composition of columns α from view I_i into columns α of I_v. This means that we begin by filling the columns of the virtual view by using columns with the same index in a view at position where the desired viewing direction α intersects the input path. This principle holds only for a very dense input sequence, in which we always have some view I_i for each α.

In practice, we have a discrete number of input images and it may happen that we need to sample a column from a view $I_{i+\Delta}$ which is not available. Then we have to use the nearest view I_i to a view $I_{i+\Delta}$. We can either use the same column α from the view I_i as we would use from the view $I_{i+\Delta}$ or we can approximate the desired viewing

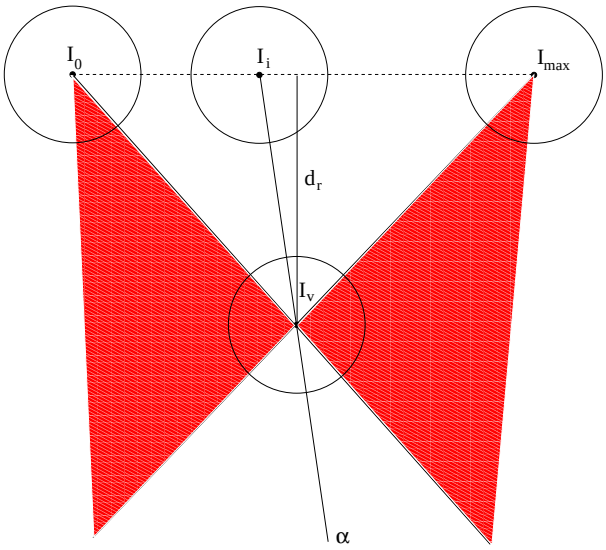

Figure 2. Omnidirectional X-slits with catadioptric images. The red shaded area marks viewing directions α which cannot be synthesized from the input set for a virtual view at this position.

direction by some column α' from the view I_i, which meets our desired light ray at some scene depth d_e, see Figure 3. The former case is equivalent to setting d_e to infinity. This holds also for the classical X-slits with narrow filed-of-view cameras.

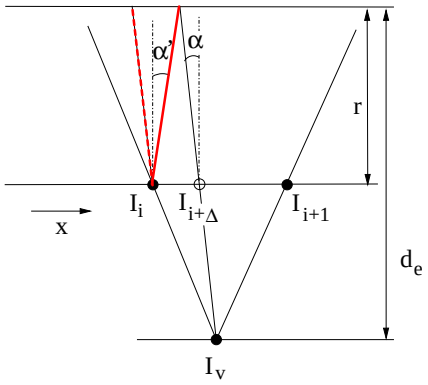

Figure 3. For scene depth at infinity, the same angle α is used from the nearest view I_i to a missing view $I_{i+\Delta}$, while for a finite scene depth, different angle α' has to be used.

What is different for panoramic images and was not discussed in [12] is that we do not always have light rays available for every α, when the position of the virtual view I_v moves away from the input line. Since the input images were acquired on a linear trajectory, we do not have light rays in the direction of motion (and light rays in the opposite direction as well) from other position than from the input trajectory. The range of α for which we do not have input data widens as we move away from the input line, see Figure 2, where is it marked by the red shaded area. We

have to mask out the respective part of the virtual image for the localization process.

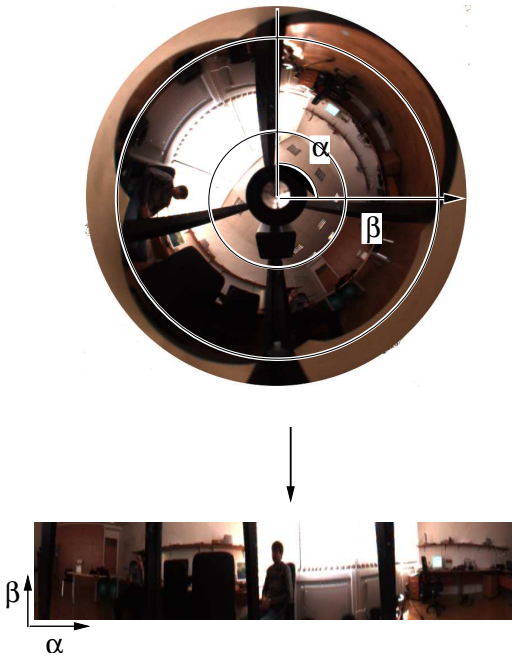

Figure 4. Transformation of a catadioptric image into a cylindrical panorama organized by angles. The two circles limit the field of view used for IBR.

We acquire panoramic images using a catadioptric sensor, when a camera observes a curved rotational symmetric mirror which reflects the scene into the image [2]. Some of these sensors have an important property called single effective viewpoint (SEV) which allows us to model the resulting sensor by a central projection [2]. We do not restrict ourselves to these kind of sensors, we only require rotational symmetry of the mirror and a rough alignment of the axis of the mirror with the optical axis of the camera.

For the following analysis and also for implementation convenience, we transfer the catadioptric images into cylindrical panoramas, where the column index determines the azimuth angle and the row index corresponds to the elevation, see Figure 4. From now on, referring to images means referring to these panoramas.

3. IBR as column ordering

The key idea in this paper is that the omnidirectional IBR presented in the previous section serves as a tool for column ordering, which allows us to handle occlusions in the scene. By occlusion we mean occlusions of static object in the scene such as chairs, not occlusion cased by dynamic or changing objects such as people. We rely on the amount of information in the omnidirectional image to cope with the latter. Because of the rotational symmetry of the catadioptric images, we do not need any calibration for this column ordering, simply measuring the azimuth angle of the respective column gives us the azimuth direction of all rays in this column. The azimuth angle has to be measured from some zero value α_0, corresponding to the direction of motion of the camera capturing the input images. We have determined this value manually, but it can be estimated from the optical flow in the images, since it corresponds to a source of the flow.

The main advantage of using IBR over feature based localization is that IBR determines the correspondences directly by this column ordering. This is a different situation from feature matching, where we detect features in the whole query image and we have to solve the problem of matching the features from the query image to features in the input images. Real scenes usually contain repetitive patters and similar objects, resulting in many false matches. Some robust technique, such as RANSAC, has to be used to resolve these ambiguities but it also requires some camera model. On the other hand, the IBR column ordering gives us a proper column order for the virtual view and we can directly try to match the columns with columns in the query view.

Moreover, by limiting the vertical field of view of the omnidirectional images to a small range covering only the 'walls' of the environment, we relax also calibration of the mapping between the light rays and radii in the omnidirectional image in the elevation direction and we can use equidistant projection model.

$$r = a\beta \ , \tag{1}$$

where r is the radius in image, β is the elevation angle and a is a model parameter.

We can also deal with non-central sensors by assuming a scene much more distant compared to the displacement of intersections with of the light rays with the axis of the mirror in a non-central sensor. This assumption will always hold for a limited FOV thus we can assume that the light rays meet in a single point, since we do not perform any triangulation of scene reconstruction where precision is required.

4. Column matching

A straightforward approach to column matching is a direct comparison of pixel values of the columns, preferably after some histogram equalization to compensate for illumination changes. For this, the columns have to be properly scaled, that is we have to know the horizon line in the images and we have to assume some scene depth d_e. Moreover, we have to store the whole input images into memory for efficient localization. The scaling function is expressed as

$$\beta' = \beta_0 + \frac{(d_e\beta)}{d_e - r} \ , \tag{2}$$

where β_0 is the horizon row index (horizon line), β is the row index of a pixel, r is the distance of the input view I_i from which we sample the column from the scene object at distance d_e from the virtual view I_v, as it is depicted in Figure 5. Note that for simplicity we always measure d_e from the actual position of the virtual view. After vertical normalization, we simply compute the correlation of the column in the vertical image with the respective column in the query image. Also note that we do not have to consider the actual height h of the object, since the relation is expressed solely in angles. The horizon line β_0 has to be determined in the images, in our case, we used manual detection, but some automatic method based on optical flow can be used in the future making use of the fact, that image points imaged to the horizon line in the image do not change their elevation while the camera moves. Figure 6 shows an example of an optical flow based horizon line determination, the features were tracked using a KLT tracker and as can be seen in detail of the source of the optical flow, we can determine features at the horizon line as well as features in the direction of motion. In this particular case, we were lucky to have a single feature determining both.

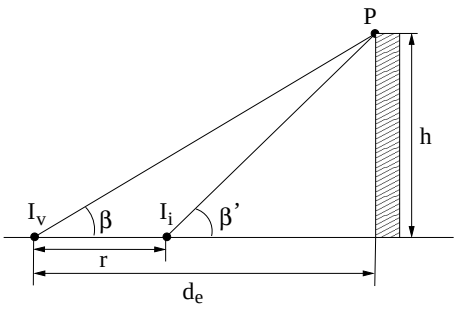

Figure 5. Vertical scaling of the X-slits images simulates forward/backward motion.

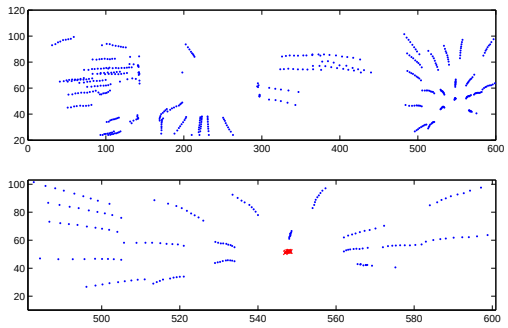

Figure 6. Top: Features tracked through every 10-th image of the input sequence. Bottom: Detail of the source of the optical flow showing a feature at a horizon and also in the direction of motion (red crosses).

5. Experimental results

We have performed a real world experiment with a mobile robot moving in an office environment. The robot was equipped with a Flea camera observing a hyperbolical mirror but no care was given to assembly of the camera–mirror rig, therefore the resulting catadioptric sensor has to be treated as non-central. The robot traversed two, almost parallel, linear trajectories and in our experiment, we used the first one as the visual representation and the second one as a source of query images. The robot was heading roughly in the same direction so we did not have to compensate for rotations of the images. Moreover, we used constant frame rate and constant speed to obtain equally spaced input and query images. This was not exactly fulfilled but due to lack of time, we did not implement support for odometry positions into our code. The distances between the input images determine the units of the environmental map, in our case 1 unit was around 2cm.

The input path led through middle of the scene and the test path was close to one of the walls, which can be considered as a difficult case for our method because the distance of the test line from the input path was about half of the input path length, making the areas of no available columns quite big in the virtual images corresponding to positions on the test line, see Figure 7.

Figure 7, top, shows one of the input images. We changed the environment by opening a cabinet and moving people and chairs to simulate real world occlusions, see Figure 7, middle. The bottom part of Figure 7 shows the closest X-slits image for the above query image. Note that the range of α where no information was available was big, thus only a small part of the scene could be rendered but it was still sufficient to localize the robot.

Figure 7. Top: one of the input images. Middle: an example of a query image. Note changes in the scene. Bottom: the closest X-slits image for the above query image. Note that only a small part of the scene could be rendered but it was sufficient to localize the robot. Only the intensity value of the pixels from the input set is used, thus the X-slits image is shown in gray scale.

First, we tested the full IBR column matching with column normalization. We selected 8 query images from the test line, equally spaced by 10 units, and run localization for each of them. Figure 8 shows the result. It can be seen that the recovered locations of the robot lie approximately on a line with similar spacing between them. Localization error is mainly due to unequal distances between the images in both the input and test datasets.

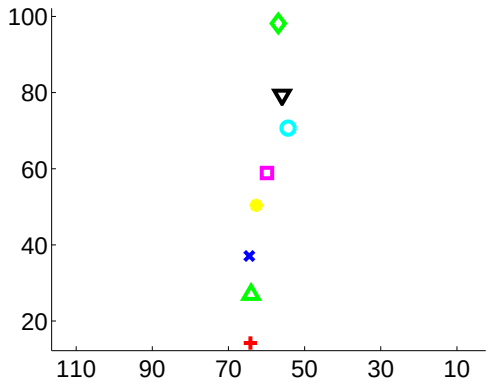

Figure 8. Localization results for query images on a linear path. The images are equally spaced by 10 units.

To investigate the precision of the localization more clearly, we used a sequence of 8 query images spaced by 1 unit and run localization for each of them, Figure 9 shows the result. Note that the locations differ from a linear trajectory by at most 1 unit and that the views were localized in a proper order, the symbols and colors in this figure correspond to the ordering the previous experiment. The localized positions are non-integer because we computed a mean value of several minimal values to cope with jitter and noise in the virtual images.

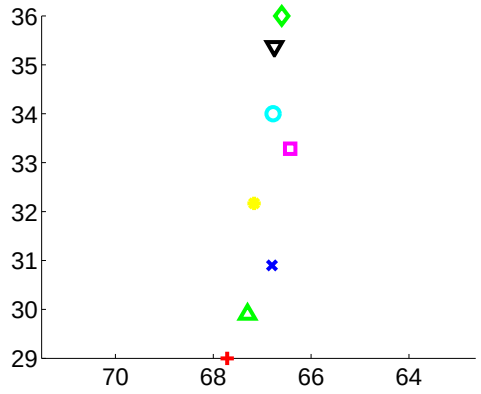

Figure 9. Localization results for dense query images on a linear path. The images are equally spaced by 1 unit.

The positions of the robot are discrete in our case, since during the X-slits rendering we use input images at discrete positions and a very small change of the position of the virtual view does not results in a change in the virtual view. Therefore, a similarity measure defined between the virtual view and the query view would be unchanged by a small change in the parameters of the X-slits rendering and therefore we cannot use conventional optimization approaches. Fortunately, our image similarity function is well behaving and we have only two parameters to search for so we can use a coarse to fine search where the global minimum can be iteratively found by brute force search over all possible values with refinement of the increment in unknowns.

Figure 10 depicts this process. We start with a big increment of the parameter values and then in the next iteration, we focus only on an area around the global minima, obtained by thresholding of the similarity values. In case the thresholding does not give us a single area, we can either choose one based on the values of similarity in the areas or we can move the robot to a new location and try another query image. In the next iteration, we refine the step of the search and obtain another minimal area and so on, until we reach a unit step in the parameter values. Note that this localization is performed only once when the robot enters an unknown environment and that the whole task can be implemented quite efficiently by remembering already computed values between iterations and by parallelization of the computation.

6. Conclusions and future work

We have presented a method for robot localization using X-slits IBR, where the localization is posed as an optimization problem where we look for a virtual view synthesized from a set of input views (the image map of the environment) which is the most similar to some query view captured at an unknown position. Parameters of the X-slits rendering then determine directly the position of the robot. We have thoroughly described the concept of panoramic X-slits rendering focusing on practical issues such as using non-SEV sensors and rough assumptions about the horizon line and scene geometry, which allows us to produce X-slits images with proper vertical normalization so that correlation can be used for column matching.

An alternative approach to correlation is a descriptor based column matching, when the column is described for example by a histogram. This reduces the amount of data dramatically, but the robustness is lower. Another way of reducing the amount of data would be to follow the idea of subspace representation of the input images, such as PCA in [1]. In our opinion, the best solution is to combine features detected in the columns with histogram of areas, where no features are present. Both the features and the histograms are matched by correlation, the feature correspondence problem is solved by the column ordering. Not

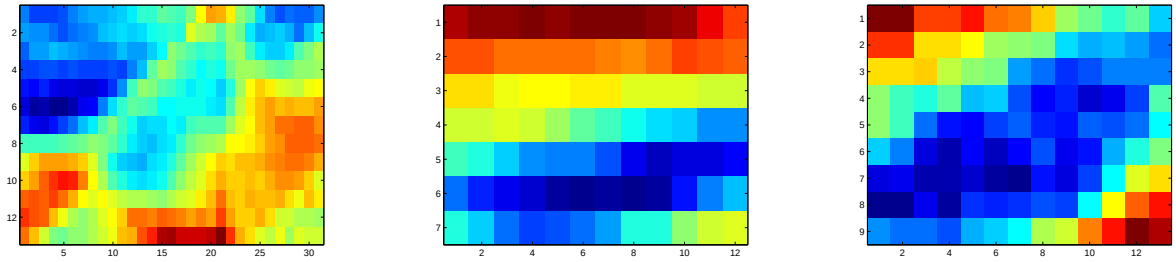

Figure 10. Coarse to fine approach to optimization. Dark blue denotes maximal values, dark red minimal. From left to right the increment of the optimized parameters was 16, 4, and 1 unit.

only that the amount of information to store will be reduced, but this approach does not require any vertical normalization and therefore we do not need to know the horizon line and the scene depth estimate. This will be our future work.

We can also try to reduce the areas, where the columns in the virtual image cannot be defined by some other input trajectory than line. For example, we can modify the rendering algorithm to use 'X' or 'L' shaped input trajectory, with the latter more convenient from the acquisition point of view; the robot can capture the whole trajectory in a single run. The former shape is, on the other hand, optimal from the rendering point of view, since it minimizes the distances of the virtual views from the input path.

Acknowledgments

Hynek Bakstein was supported by the STINT Foundation under project Dur IG2003-2 062 and by the EC project FP6-IST-027787 DIRAC and by The Czech Government under the research program MSM6840770038 and by MSMT under project 9-06-17. Aleš Leonardis was supported by: Research program P2-0214 (RS), EU FP6-004250-IP project CoSy, and EU FP6-511051-2 project MOBVIS and CZ-SI S&T Cooperation Programme. Any opinions expressed in this paper do not necessarily reflect the views of the European Community. The Community is not liable for any use that may be made of the information contained herein.

References

[1] M. Artač, H. Bakstein, M. Jogan, and A. Leonardis. Panoramic volumes for robot localization. In C. P. Chen, P. X. Liu, and Z. Wang, editors, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2005 (IROS'05)*, pages 3776–3782, 2600 Anderson Street, WI 53704, Madison, USA, Aug. 2005. Omnipress. dvd only. 2, 5

[2] S. Baker and S. K. Nayar. A theory of catadioptric image formation. In *Proc. of the International Conference on Computer Vision (ICCV'98), Bombay*, volume 1, January 1998. 3

[3] M. Jogan and A. Leonardis. Robust localization using an omnidirectional appearance-based subspace model of environment. *Robotics and Autonomous Systems, Elsevier Science*, 45(1):51–72, 2003. 1

[4] A. Levin and R. Szeliski. Visual odometry and map correlation. In *Proceedings of Computer Vision and Pattern Recognition Conference (CVPR)*, pages 611–618, 2004. 1

[5] A. Makadia and K. Daniilidis. Rotation recovery from spherical images without correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1170–1175, 2006. 2

[6] T. Pajdla and V. Hlavac. Zero phase representation of panoramic images for image based localization. In *Comp. Analysis of Images and Patterns*, pages 550–557, 1999. 1

[7] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotic Research*, 8(21):735–760, August 2002. 1

[8] S. Se, D. G. Lowe, and J. J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005. 1

[9] H.-Y. Shum and S. B. Kang. A review of image-based rendering techniques. In *IEEE/SPIE Visual Communications and Image Processing (VCIP) 2000*, pages 2–13, June 2000. 1

[10] C. Silpa-Anan and R. Hartley. Visual localization and loopback detection with a high resolution omnidirectional camera. In *Proceedings of the Omnivis workshop*, 2005. 1

[11] Y. Yagi, K. Imai, and M. Yachida. Iconic memory-based omnidirectional route panorama navigation. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 564–570, Taipei, Taiwan, September 2003. 2

[12] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: The crossed-slits projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(6):741–754, June 2003. 2