ARToolkit Applied to Panoramic Vision for Robot Navigation

Mark Fiala
Computational Video Group, National Research Council, Ottawa, Canada K1A 0R6
mark.fiala@nrc-cnrc.gc.ca.

Abstract

A vision based navigation system is presented for determining a mobile robot's position and orientation using panoramic imagery. An omni-directional image sensor mounted on the robot is useful in obtaining a 360° *field of view, permitting navigational markers from all sides to be viewed simultaneously. In this novel approach, navigational markers are recognized using software from the popular* ARToolkit *software package. These marker patterns are arrayed in the environment and recognized in quasi-perspective projection views warped from the panoramic image. A novel way is shown for generating these perspective projection images from the non-perspective image seen by the panoramic camera, without relying on the* Single Viewpoint Criteria *which greatly restricts the camera design. An* a priori *model is used to predict the marker positions, from which quasi-perspective views are generated. The* ARToolkit *software operates on these views and identifies and locates the markers' directions from which the robot's position is updated. Robust performance is achieved by the uniqueness of the markers as landmark features. Experiments are shown with synthetic imagery.* **Keywords:** Robot Navigation, Panoramic, Omnidirectional Vision

1 Introduction

One fundamental component for an autonomous mobile robotic platform is to determine it's position and orientation with respect to it's environment. Example systems use sonar sensors, motor odometry and radio beacons. A passive vision-based system would be very advantageous, and increase the practical utility and scalability of mobile robotics. Hager and Rasmussen define a framework for robot navigation using standard perspective cameras [9, 11]. If this vision system was panoramic, objects all around the robot could be used for finding and updating the position estimate.

This work follows on previous research [8] where a mobile robot navigation system was built using corners in the environment as navigational landmarks. This was achieved using an un-restrictive camera geometry that avoids having to satisfy the *Single Viewpoint Criteria* through use of the *Panoramic Hough Transform* [5]. This previous research [8] relied on the successful tracking of corners which are not unique features. Similar to many tracking problems, the frame rate has to be high enough to use a small enough search window such that a given corner is not confused with other corners. Even with the improvement of tracking lines and defining corners as junctions, as opposed to using a corner detector, the system could still under certain conditions falsely identify these landmark corners leading to an erroneous robot position estimate. The shortcomings are due to the non-uniqueness of the landmark features.

ARToolkit [10] is a freely available and widely used software package intended for *Augmented Reality* but is applied here to the task of robot navigation. The software detects the presence of a pre-programmed rectangular marker in an image, and outputs the marker number and location of the four marker corner points. These corner points are typically used to then calculate some camera pose parameters to augment virtual objects onto the image, but the center of them can be used instead as a landmark point. Using the *ARToolkit* markers and software allows the robust identification of unique landmarks, and so can be used for robot navigation in place of environment corners.

However, *ARToolkit* requires traditional perspective projection images and not the curved projection that 360^o panoramic cameras provide. Imagery from other projections can be used if equivalent perspective projections can be created through image warping. To achieve this, the camera must possess the Single Viewpoint Criteria which states that all light captured in the image pass through a (real or virtual) single point.

If the panoramic camera is a *catadioptric* system (containing both mirror and lens elements), then only two restrictive geometries can be used. This creates a cumbersome and expensive panoramic camera. However, simpler catadioptric systems can be used if approximate quasi-perspective projection views can suffice using non-SVP panoramic cameras. This paper shows how to create these views using a camera design with simple components, where these views are sufficient to allow *ARToolkit* to perform the marker recognition task.

This system would be useful where the minor modifications to the environment of mounting a few planar pattern markers is acceptable. The method described would allow for automatic discovery and mapping of the environment, but only a system assuming an *a priori* map is described herein.

2 Panoramic Image Sensor

Omni-directional viewing would require many standard narrow field of view cameras, or one panoramic camera. A *catadioptric* optical system is one that uses both lens and mirror components in the optical path, and can be used to capture a 360° field of view around a mobile robot. An example is shown in Fig.1a. If a convex, radially symmetric mirror, lens(es) or pinhole and a planar image plane are all mounted along one vertical axis, then a panoramic view can be captured as shown in Fig.1b. Such a system has the advantages of processing only one image, and with this image being continuous, not having to deal with discontinuities at the boundaries of view as a ring of conventional cameras would introduce.

Equivalent perspective projection images can be extracted from this panoramic image only if the catadioptric geometry satisfies the *Single Viewpoint Criteria*. The model of a perspective pinhole projection cannot be used in panoramic catadioptric imagery unless a single view point (virtual perspective point) exists, where light rays are captured whose direction all converge at a virtual perspective point (usually inside the mirror). Baker and Nayar [1] have shown that this is only possible if the mirror has a parabolic or hyperbolic profile. With all other mirror profiles this is not the case and the direction of captured light rays have no such convergence point, and the optical system is said to be *Non-Single Viewpoint* (non-SVP). To satisfy the Single Viewpoint Criteria with parabolic and hyperbolic profiles certain relative dimensions must be followed, and an expensive telecentric lens used with the parabolic profile.

However many other mirror profiles are desirable for several reasons, an example being circular (spherical mirrors) for the ease of their manufacture. Other useful panoramic mirror profiles are designed to shape the density of image resolution as a function of elevation, either to evenly distribute the image resolution throughout a desired range [3], or to improve resolution at certain elevations. Derrien [4] demonstrated various advantages to relaxing the single viewpoint constraint for panoramic catadioptric camera design.

Issues with using non-SVP panoramic optics are addressed in [5], specifically that of the absence of the preservation of the straightness of line features, a phenomenon that benefit traditional perspective image analysis. The *Panoramic Hough Transform* is a tool originally designed to aid in line detection without restricting the mirror geometry to that of the special mirror profiles, dimensions and non-standard lenses.

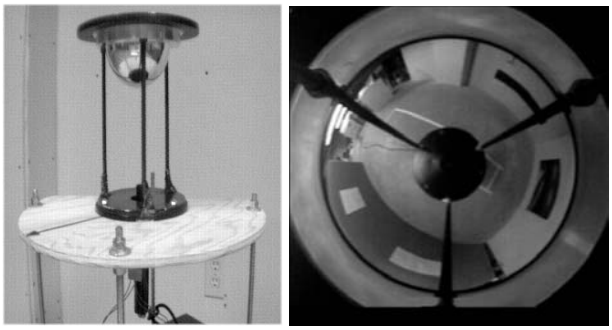

Figure 1: *(A) Panoramic Imaging system using Catadioptric optics (shown mounted on a mobile platform), and (B) a sample panoramic image.*

The theory and look-up tables used for this transform can also be used to extract a quasi-perspective view from the panoramic image. Using this approach allows the contour of the projection of horizontal lines to be located in the panoramic image. With a given center angle, each row of the quasi-perspective view can be filled with pixels lying along the corresponding contour.

3 ARToolkit Markers

Planar patterns can provide unique landmarks. Locating and identifying simple planar patterns is used by several photogrammetry and augmented reality systems. The *HOM, IGD, SCR* systems use a small grid of squares or circles providing identification and relative pose information. Zhang [12] performs a survey of these three and *ARToolkit* with respect to processsing time and identification, image position accuracy with respect to viewing angle and distance. *ARToolkit* was chosen for the robot navigation system due to it's available source code and ubiquity of use.

Some of the original *ARToolkit* markers are shown below in Fig.2 and were used in the experiments. Owen [2] demonstrates that these original patterns inside the black border are not necessarily optimal, and proposes others (the original ones are used in this paper).

Figure 2: *Some ARToolkit Markers.*

4 Triangulating Location From Landmarks

The landmark tracking function provides a set of detected landmarks and their azimuth angle. If the the mobile robot is constrained to movement on a horizontal plane, the elevation of landmarks is used in the tracking, but only the angle is required to find position. Thus the result of landmark tracking need only be a set of landmark labels and their θ_l angles.

If the robot's camera orientation angle θ_c is assumed, then the position must lie along the line drawn from a detected landmark's world model position, along the azimuth angle $\theta_l - \theta_c$. This line can be described by $Ax + By + C = 0$ where $a = sin(\theta_l - \theta_c)$, $b = sin(\theta_l - \theta_c)$ and $c = 1$. The camera location can be found by finding the convergence of all such lines. Assuming equal confidence for all θ_l angles, the camera position (X_c, Y_c) can be found by the method of least squares (Eqn. 1), where the quantity minimized is the perpendicular distance from the camera position to all the lines (Eqn. 1).

$$X_c = \frac{-\sum b^2 \sum ac + \sum bc \sum ab}{\sum a^2 \sum b^2 - (\sum ab)^2}$$
$$Y_c = -\frac{\sum a^2 \sum bc - \sum ac \sum ab}{\sum a^2 \sum b^2 - (\sum ab)^2} \quad (1)$$

$$\sum dist^2 = X_c^2 \sum a^2 + 2X_c Y_c \sum ab + \sum c^2$$
$$+2X_c \sum ac + Y_c^2 \sum b^2 + 2Y_c \sum bc \quad (2)$$

The above assumes a known θ_c. It was found to be sufficient to calculate θ_c and (X_c, Y_c) independently as that a θ_c value within $+/-45^{\cdot}$ of the correct value yields almost the same position. Newton-raphson iteration with θ_c to find a minimum of Eqn. 2 produces θ_c, which is then used in Eqn. 1 to find (X_c, Y_c).

5 Finding Projections of Horizontal Lines in Panoramic Imagery

Finding the projection of horizontal and vertical lines allows a mapping between image pixels in the panoramic image and points on a vertical planar surface in the environment. This vertical planar surface can instead represent the image plane of an equivalent perspective projection camera. Thus a quasi-perspective warp can be created by mapping back from each point on a hypothetical vertical plane to the panoramic image. This is only an approximation since the light captured by a non-SVP camera has not passed through a single point, however it produces images close enough for this marker recognition application.

If a catadioptric panoramic camera is positioned with it's central axis vertical, then vertical lines project simply to radial lines. However, horizontal lines in the environment project to curved contours which depend on the mirror profile. This section describes how the contours for the complex horizontal line projections are found.

The projection of straight horizontal lines in the imagery provided by SVP and non-SVP catadioptric panoramic image sensors was presented in [5]. The relative position of a scene point $P(x, y, z)$ to the camera can be expressed as lying along a horizontal line whose closest approach to the camera axis occurs at a direction of θ_{main}. This line is defined by the direction θ_{main}, the distance D_{main} and height Z_{main}. $P(x, y, z)$ can be defined by a 4^{th} parameter $d\theta$ relative to θ_{main}. This is shown in Fig.3.

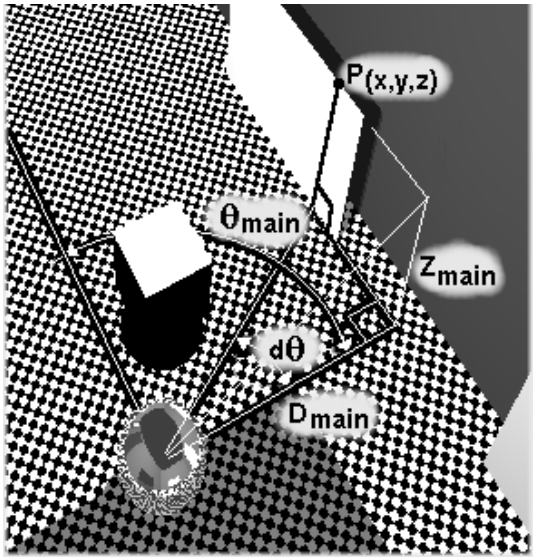

Figure 3: *Formation of image plane contour from a 3D horizontal line. Scene point $P(x, y, z)$ can be represented by angle $d\theta$ along a horizontal line defined by θ_{main}, D_{main} and Z_{main}.*

The horizontal line that contains $P(x, y, z)$ has three parameters; θ_{main}, distance D_{main} and height Z_{main}. This projects onto a curved line on the image plane defined by θ_{main} and R_{main}. Due to the loss of depth information, a single parameter R_{main} is a function of D_{main} and height Z_{main} (Eqn. 5). This function depends on the mirror profile, the focal length and the distance from the lens (or pinhole) to the mirror. R_{main} and θ_{main} are enough to define the projection of the horizontal line containing $P(x, y, z)$. Fig.4 shows how the scene point $P(x, y, z)$ will appear in the image plane as $P(u, v)$. θ_{main} and $d\theta$ are preserved in the projection.

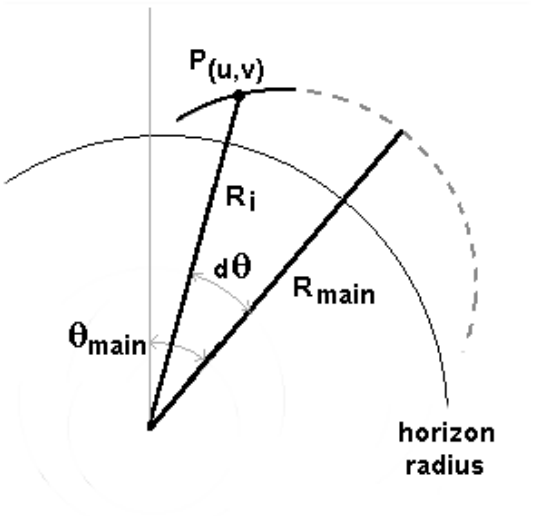

Figure 4: *The projection onto the image plane can be represented by* $R_i, \theta_{main} + d\theta$.

The parameter $d\theta$ is necessary to uniquely define a $P(x, y, z)$ point along the 3D line or image point $P(u, v)$ on the image contour. Due to the radial symmetry, the shape of the projection of a horizontal line is only a function of R_{main} and $d\theta$, and given in polar coordinates in Eqn. 4. Eqns. 3 and Eqn. 4 are defined as the *Panoramic Hough Transform* in [5, 6, 7] and are used in a hough transform method to find the possible projections of horizontal lines from a set of edge points in those publications.

$$R_{main} = F(R_i, d\theta) \quad (3)$$

$$R_i = F^{-1}(R_{main}, d\theta) \quad (4)$$

$$R_{main} = E(\frac{D_{main}}{cos(0)}, Z_{main}) = F(R_{main}, 0) \quad (5)$$

To find a point $P(u, v)$ along a contour described by R_{main} and θ_{main} corresponding to $d\theta$, Eqn. 4 is employed. $P(u, v)$ can be expressed in polar coordinates as $P(R_i, \theta_i) = P(R_i, \theta_{main} + d\theta)$.

The function $R_i = F^{-1}(R_{main}, d\theta)$ is a function of the mirror profile and the function $E(distance, height)$ from Eqn. 5. $R_i = U = E(P_x, P_y)$ from Fig.5. Examples of several conic mirror profiles are given (Eqn. 6).

$$\begin{aligned} y &= G(x) \\ x^2 + y^2 &= r^2 \quad CircularProfile \\ y &= a \cdot x^2 + b \quad ParabolicProfile \\ a \cdot x^2 - b \cdot y^2 &= r^2 \quad HyperbolicProfile \end{aligned} \quad (6)$$

A spherical mirror has a circular profile, shown in Fig.5 The function $R_i = F^{-1}(R_{main}, d\theta)$ can be determined by analyzing a planar slice of Fig.3. The panoramic camera can be divided into a mirror and a conventional *dioptric* (containing only lens elements in the optical path).

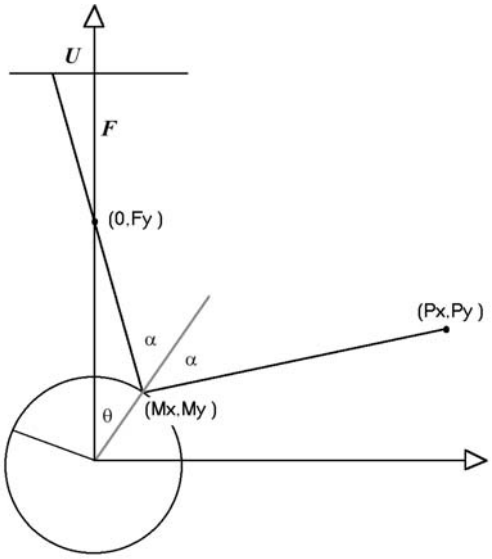

Figure 5: *Modeling the projection of a point onto the image plane after reflection off of a spherical mirror of radius* R. P_x, P_y *represent* D_{main}, Z_{main} *from Fig.3 and the point of reflection on the mirror has coordinates* (m_x, m_y). *The dioptric (lensed) camera is located at* $y = f_y$ *above the mirror.*

Our derivation concentrates on spherical mirrors, whose normal vector at any point it given by Eqn. 7. The perfect reflection condition of the angle of incidence being equal to the angle of reflection (Eqn. 8).

$$\overrightarrow{N} = \frac{1}{r} \begin{bmatrix} m_x \\ m_y \end{bmatrix} \quad (7)$$

$$\alpha_i = \alpha_r \quad (8)$$

These angles are calculated by the dot product between the unit direction vector from the point being imaged (p_x, p_y) (Eqn. 9) to the mirror point (m_x, m_y) and the unit direction vector from the focal point $(0, f_y)$ to this same mirror point (Eqn. 10). The corresponding point on the image plane is given by the simple perspective Eqn. (Eqn. 14). Eqn.s 6, 7, 8 and 9 through 14 are sufficient to constrain the geometry and provide a solution mapping a point in space to a point on the image plane.

$$\overrightarrow{V_i} = \frac{1}{((p_x - m_x)^2 + (p_y - m_y)^2} \begin{bmatrix} p_x - m_x \\ p_y - m_y \end{bmatrix} \quad (9)$$

$$\overrightarrow{V_r} = \frac{1}{(m_x^2 + (f_y - m_y)^2} \begin{bmatrix} -m_x \\ f_y - m_y \end{bmatrix} \quad (10)$$

$$MirrorNormalVector\overrightarrow{N} = \frac{1}{r} \begin{bmatrix} m_x \\ m_y \end{bmatrix} \quad (11)$$

$$cos(\alpha_{i)} = \overrightarrow{V_i} \cdot \overrightarrow{N} \quad (12)$$

$$cos(\alpha_r) = \overrightarrow{V_r} \cdot \overrightarrow{N} \quad (13)$$

$$\frac{u}{f_y} = \frac{m_x}{m_y - f_y} \quad (14)$$

The equation of the radial profile affects this mapping, a closed form for the projection of a horizontal line after reflecting off a spherical mirror (as a function of a parametric line scalar) could not be found, and numerical methods were required to find this mapping.

The function $R_i = F^{-1}(R_{main}, d\theta)$ is thus found for the mirror profile. Pratically it was found numerically and entered into a 2D lookup table where R_i was found as a result of indices R_{main} and $d\theta$. The lookup table is displayed graphically below in 6.

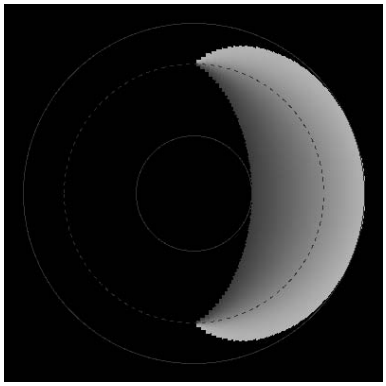

Figure 6: *Graphical display of lookup table. R_i is shown as an intensity value, the minimum, maximum and horizon radii are shown.*

6 Creating Quasi-Perspective Views from non-SVP Panoramic Images

A quasi-perspective view of width w can be found by mapping points from the panoramic image for a given angle and angular width using Eqn. 4. Each horizontal pixel position i is converted to a $d\theta$ angle according to $d\theta = tan^{-1}(\frac{i-w/2}{w})$. The vertical pixel position j maps to R_{main} and is used to calculate R_i.

This can be visualised as reprojecting from the image back to a vertical rectangle. Starting with each pixel point on the dioptric camera's image plane, each beam passes though the focal point, reflects off the mirror and colours the point where the beam hits the vertical rectangle. The vertical plane is perpendicular to and centered about a given angle θ_{main}. Then $R_i = F^{-1}(R_{main}, d\theta)$ is found, and the image point $(R_i cos(\theta_{main}), R_i sin(\theta_{main}))$ corresponding to (i, j) obtained. The image pixel can either be sampled, or some operation such as bilinear interpolation employed as is done in the experiments below.

The following image (Fig.7) is the first frame of the image sequence used for synthetic imagery experiment. Quasi-perspective warps (Fig.8) are found for $\theta_{main} = 180^o, 350^o, 80^o$ and 90^o (measured clockwise from the top).

Figure 7: *The view seen by the panoramic camera of a synthetic environment with* ARToolkit *markers.*

7 Experiment

A synthetic sequence was rendered to create simulated video from a robot carrying a non-SVP catadioptric camera of spherical profile. The start location and *a priori* model was provided and the system successfully tracked the robot motion using only the image sequence. 66 frames were generated synthetically using the freely available *Povray* ray tracing package. The route with 17 ARToolkit markers is shown in Fig.9. A top view of the camera and marker filled environment of the first frame is shown in Fig.10.

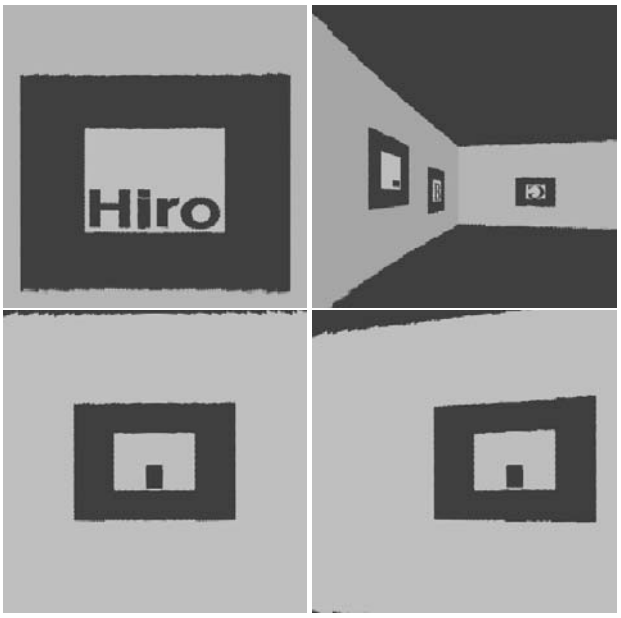

Figure 8: *Four Quasi-Perspective Views created from Fig.7.*

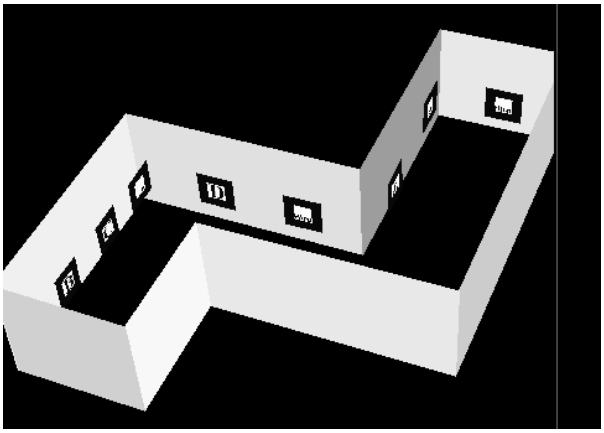

Figure 9: *Environment used to create image sequence. 17 ARToolkit markers are used as landmarks.*

The procedure performed on each image frame is shown in Fig. 11. Quasi-perspective view images are warped from the panoramic image, input to ARToolkit and the successfully detected markers used as landmarks to update the robot's position.

A plot of the recovered trajectory is shown below in Fig.12. All original and result images can be viewed online. The results can be viewed graphically at [1].

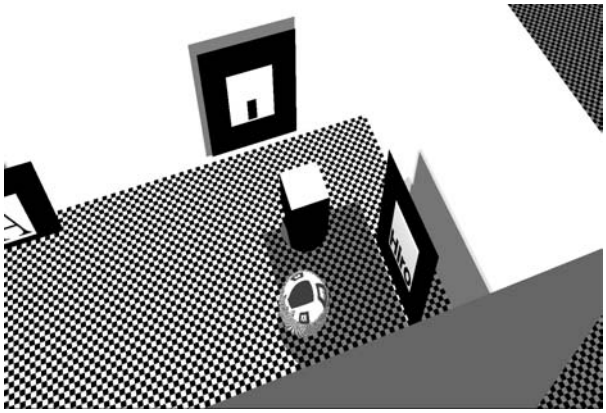

Figure 10: *Top view of synthetic scene. The panoramic camera is composed of a conventional* dioptric *camera looking down on a rounded mirror. In this experiment the mirror is spherical (circular profile). The view captured by the camera is shown in Fig. 7.*

The synthetically generated sequence successfully recovered the camera trajectory, with a standard deviation of 0.54 units, a typical error of 0.8 percent over the average landmark range of 70 units. These results are very close to the results obtained in the previous research [8] where the same synthetic scene was used using a polygon vertices instead of planar marker patterns. In that experiment an error of 0.4 units was found over the same range. Repeating the error estimate calculation from Eqn. 15, this indicates just less than a one pixel accuracy on corner location detection. However, more important than the position accuracy is the absence of outlier points due to the uniqueness of the marker patterns.

It should be noted that the Eqns. 1,2 rely on a minimum of three landmark points being found. In this experiment there were always at least three ARToolkit markers in view, but for a real world system different methods would have to be used for times when less than three were detected. A measured angle between two landmarks reduces the possible locations from three dimensions to one, but not to a point. One way to handle this could be to find the point on this locus closest to the estimated position.

A rough expected error estimate can be made from an

[1] /http://www.cv.iit.nrc.ca/ fiala/

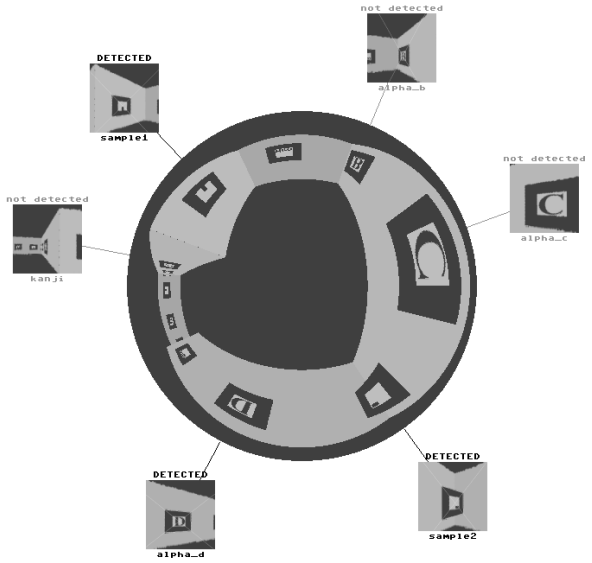

Figure 11: *The process demonstrated for frame 55/66 of the synthetic sequence. The center is the image captured by the panoramic camera on the robot, and the radial lines indicate the direction of predicted markers. The smaller images at the end of the radial lines are the view images (quasi-perspective warps) extracted from the main image. These 6 views are input to the ARToolkit code which in this example locates 3 markers. The text over each image display 'DETECTED' if the marker was both detected and correctly identified. Only the markers that satisfy both conditions are used as landmarks for triangulating the robot's location. The landmark angle is adjusted for where ARToolkit detected the marker in the view image. This frame was chosen as a worst case, where only the minimum 3 markers are detected, typically more were successfully detected to use as landmarks.*

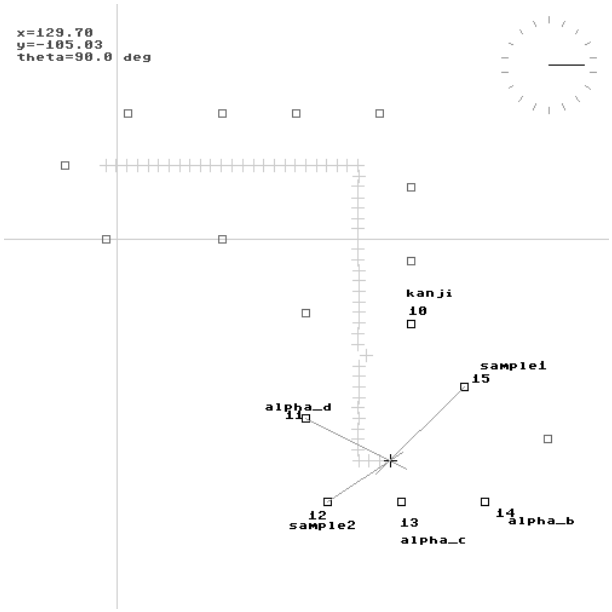

Figure 12: *The recovered trajectory at frame 55/66 of the synthetic sequence. The text and compass graphic at the top are the camera position for this frame. The crosshairs show the current and previous 54 triangulated positions. This image is a top view schematic showing the ARToolkit markers as small squares, with the labelled ones indicating the markers expected to be in view at the estimated current location. The intersecting line segments indicate the current location triangulated from the detected markers. In this case, only three markers were detected.*

assumption of a marker detection accuracy of $Error_{det}$ pixels. ARToolkit typically achieves sub-pixel accuracy in the virtual perspective views, but this cannot translate into an accuracy greater than 1 pixel in the original panoramic image. The angular error can be estimated by the circumference out at the *horizon radius* rad_h. The camera's position is indeterminate to about $Error_x = 2\pi\frac{1}{2\pi rad_h} = \frac{1}{rad_h}$ of the average distance to the landmark. If we assume two corner landmarks at 90° angle then the best expected error would be a region of uncertainty of:

$$Error_{dist} = \sqrt{2}X_{error} = \sqrt{2}\frac{Error_{det}}{rad_h} \qquad (15)$$

8 Discussion and Future Work

The system was demonstrated only with simulated imagery, but it is expected to function well given the performance of the similar previous work that worked in a real mobile robot implementation using less distinct landmarks. Future work would clearly be to implement this on a real mobile robot and assess real world robustness as in [8].

This paper is an extension of the work from [8] where synthetic and real image sequences were used, and a working near real-time prototype robot system created. The real-time system was not very robust to sudden jerking movements, as expected since it was tracking non-unique landmarks. It is expected that this new ARToolkit based system will be more robust to sudden loss of tracking as that the landmarks are very unique.

The virtual perspective views in this experiment were created according to viewing angles predicted by the previous location. In the event that the position is lost, the system could go to a search mode where virtual perspective views are generated every 45^o or so and examined for markers. Depending on the reuse of the marker patterns, the potential positions could be quite unique. In this way the system could be made to robustly recover from sudden jerking motions, or sudden insertion into the environment without being told a start position.

The set of unique markers, and where they are placed will influence the uniqueness of possible views. This experiment used 9 marker patterns and reused them for the 14 markers. The set of marker patterns can be increased up to the limit that ARToolkit, or equivalent marker detection system), can reliably uniquely identify them. Owen [2] analyses the similarity between patterns and proposes a system based on frequency components to create a set of patterns according to a given resolution. Since the robot is assumed to be translating along a horizontal plane, a given set of marker patterns can be reused 4 to 8 times by merely rotating the marker placement (by 90^o or 45^o respectively). The set can be effectively expanded this way by encoding the marker angle in some marker ID.

The basic set of ARToolkit markers are not very ideal, in the experiment, the 'A' pattern was frequently confused with the 'kanji' pattern. The 'sample1' and 'sample2' patterns (single black bar inside blank white interior) were identified the most robustly. A practical system would likely use a better, more distinct marker set as described earlier.

The experiment worked well with a small image size, 500 by 500 pixels with the usable area less than 400 pixels across. This size was chosen because it is comparable to that of a standard NTSC video camera. More reliable ARToolkit performance would likely occur at a larger resolution, such as 1000 by 1000 or 1500 by 1500 pixels.

The search mode described above for tracking loss recovery can also be used for map making. If the system started with at least markers of known location, the environment could be examined at unmapped angles for markers whose location could be triangulated as the robot translated using the known markers for positioning.

Planar pattern detection and localisation methods, such as the system in ARToolkit used by this research, provide more information than just a center point. An estimated distance and relative pose to a single marker can be extracted from the four corner points detected in the same way that the camera pose is determined for rendering virtual objects (the original intent of the ARToolkit). In this way the robot position could be determined, albeit it with likely lower accuracy, with only one or two markers.

Finally, the marker detection system could be integrated with wheel odometry information so that markers could be placed more sparsely. This paper assumed no such information was available and that at least three markers had to be visible at all times. With odometry the markers could be placed very sparsely in just a few locations where the robot's accumulating position error could be reset.

9 Conclusions

A vision based navigation system was presented for determining a mobile robot's position and orientation using panoramic imagery. A paradigm was introduced of extracting equivalent perspective views in chosen directions from panoramic images to more readily locate landmarks. Unique planar patterns could be used for these landmarks, this paper used the popular *ARToolkit* system. The markers could be attached to vertical surfaces in the robot's environment and identified from these virtual views.

The *Single Viewpoint Criteria* was explained and it's restricting effect on the design of a lens and mirror (catadioptric) panoramic camera. A way was shown to use *non-SVP* catadioptric cameras to allow for simple, more compact and less expensive panoramic cameras to be used with this navigation paradigm.

A synthetic experiment was performed with good results and a future plan is outlined to apply this to a real implemented system capable of recovery from sudden unexpected motion that causes unavoidable position loss with previous tracking based systems.

It is hoped that this paper helps to demonstrate the feasibility of robust, inexpensive, passive vision based mobile robot navigation systems with panoramic optics.

References

[1] S. Baker and S. Nayar. A theory of catadioptric image formation. In *IEEE ICCC Conference*, pages 392–197, 1998.

[2] F. Xiao C. Owen and P. Middlin. What is the best fiducial? In *First IEEE International Augmented Reality Toolkit Workshop (at ISMAR)*, Sep 2002.

[3] T. Conroy and J. Moore. Resolution invariant surfaces for panoramic vision systems. In *IEEE ICCV Conference*, pages 392–397, 1999.

[4] S. Derrien and K. Konolige. Approximating a single viewpoint in panoramic imaging devices. In *IEEE Proc. Omnidirectional Vision 2000*, pages 85–90, August 2000.

[5] M. Fiala and A. Basu. Hough transform for feature detection in panoramic images. In *Pattern Recognition Letters*, volume 23, 2002.

[6] M. Fiala and A. Basu. Line segment extraction in panoramic images. In *Journal of WSCG*, volume 10, pages 179–186, 2002.

[7] M. Fiala and A. Basu. Panoramic stereo reconstruction using non-svp optics. In *International Conference on Pattern Recognition (ICPR'02)*, volume 4, pages 27–30, Aug 2002.

[8] M. Fiala and A. Basu. Robot navigation using panoramic landmark tracking. In *Proc. of Vision Interface*, pages 117–124, May 2002.

[9] E. Yeh G. Hager, D. Kriegman and C. Rasmussen. Image-based prediction of landmark features for mobile robot navigation. In *IEEE Robotics and Automation*, volume 2, pages 1040–1046, 1997.

[10] I. Poupyrev H. Kato, M. Billinghurst. *ARToolkit User Manual, Version 2.33*. Human Interface Technology Lab, University of Washington, 2000.

[11] G. Hager and C. Rasmussen. Robot navigation using image sequences. In *AAAI Conference on Artificial Intelligence*, pages 938–943, 1996.

[12] S. Fronz X. Zhang and N. Navab. Visual marker detection and decoding in ar sytems: A comparative study. In *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 97–106, Sep 2002.