HAVE 2005 - IEEE International Workshop on
Haptic Audio Visual Environments and their Applications
Ottawa, Ontario, Canada, 1–2 October, 2005

Automatic Alignment and Graph Map Building of Panoramas

Mark Fiala
National Research Council of Canada, NRC 1200 Montreal RD, Ottawa, Canada K1A-0R6
e-mail: mark.fiala@nrc-cnrc.gc.ca

Gerhard Roth
National Research Council of Canada, NRC 1200 Montreal RD, Ottawa, Canada K1A-0R6
e-mail: gerhard.roth@nrc-cnrc.gc.ca

Abstract – *Panoramic cameras can capture a* 360° *view from a point providing new capabilities for multimedia, tele-presence and robotic applications. For example, virtual walk-throughs of an environment can be created from a sequence of panoramic images, where perspective views are created according to a user's position and view direction. For this and other applications, the panoramic images need to be aligned to one another and a topological or metric map created. An automatic method to achieve this would remove a lot of tedious preparations for multimedia systems and enable robotic positioning systems. This paper presents three methods to address these problems; finding the relative orientation between panoramas, using the essential matrix is created to determine the relative rotation and translation direction, and an image search based algorithm to detect when the camera path crosses over itself for creating a topological map. The* SIFT *feature detector is used to find correspondences between panoramic images. Experimental results are shown for determining the rotation and cross-overs.*

Keywords – *panoramic SIFT essential matrix.*

I. INTRODUCTION

Panoramic image sensors, capable of capturing a large section of a spherical view, open new possibilities over conventional narrow field of view cameras. Panoramic cameras have been around for several years, however only recently have ones been available with a sufficient resolution to be used in multimedia applications. One application is for virtual walk-throughs [7] and tele-presence [6] where an HMD presents a perspective view in the direction sensed by an orientation sensor (Fig.1). This gives the user the ability to look around a pre-captured or virtual scene and naturally see the view they would see if present at the remote, recorded, or virtual location.

Recently, multi-sensor cameras have arrived which enable a high enough pixel density per solid angle to allow virtual perspective views to be created with a resolution that human users expect to see. These cameras also enable new 3D automatic modeling and robotic applications.

This paper is part of the *NAVIRE* project at the University of Ottawa [1], it is motivated by the need to automatically align panoramas for use in the *cube explorer* system. The multi-sensor panoramic camera can quickly capture many images, however it is time consuming to align them and assign them to a map. While a metric cartesian description of panorama points may be desirable, it is sufficient in many cases to have only the orientation (rotation matrix or azimuth angle relative to a global reference) and a translation direction between panoramas. Also, a topological description of panorama locations is needed so that the user can navigate along these paths experiencing this captured media. If the translation direction can be found, a ”quasi-cartesian” map can be made, a topological graph with known directions between nodes. Initially the panorama sequence is merely in linear list, this needs to be converted to a graph where intersections are automatically detected.

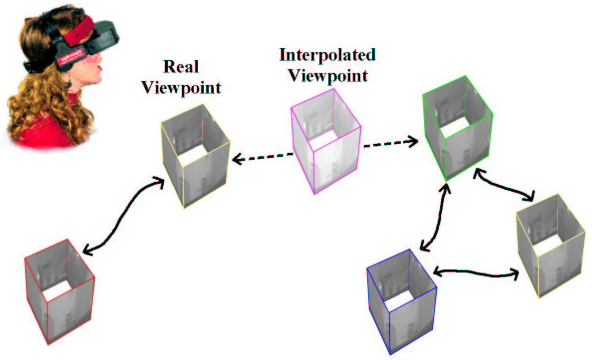

Fig. 1. *Immersive viewing of panoramas. User can look around each panorama in the set, and move between them. The viewpoint is either at locations where the panoramas were captured or from panoramas interpolated with image based rendering.*

We address this by defining it as two problems; 1) for a pair of panoramas find the relative rotation, or rotation and translation direction, and 2) detect from a sequence the event of the camera crossing its own path. A solution to the former is provided by finding either a rotation matrix or essential matrix between panorama images, and the latter is solved by combining image search techniques to detect similar images..

Both solutions use point correspondences between panoramic images provided by natural feature detectors. These methods are useful for 3D modeling, such as preparing a large image set for bundle adjustment by breaking it into connected units, and for mobile robot navigation.

The panoramas are encoded in cube form, with 6 sides each of which is a perspective projection.

Structure from motion attempts to find 3D structure and camera positions from a sequence of images of an unknown scene captured from unknown positions. This relies on being able to find correspondences between images and rigid motion assumptions [10]. Sato [14] and others [8] have demonstrated the use of natural features, such as *SIFT* features to simultaneously find the camera extrinsic parameters and 3D locations of the features in panoramic images.

Herein we present an approach to find information necessary for the creation of an aligned topological map without having to create 3D information as in structure from motion techniques.

A. *Panoramic Cameras*

A *panoramic image* is a sampling of *plenoptic function* from a single (or approximately single) point in space and because of its wide view needs to be treated differently than standard perspective images. The challenges of working with this alternate image type is offset by the benefits afforded by being able to see in all azimuth directions simultaneously.

Previous work such as [7], [3], [2], [9] and many others use a single image sensor (CCD or CMOS 2D array) and a combination of lenses and mirrors (a *catadioptric* system) to capture the light from all azimuth angles and a range of elevation angles. The GRASP webpage [1] details many of these systems. Fiala showed the limited resolution for several catadioptric configurations [6], the whole panorama is contained in an annular region in a single image, the image resolution is poor even with a high resolution modern digital video system.

The use of multiple image sensors provides a much better result, a higher resolution can be achieved with a significantly simpler optical system if multiple sensors are employed. The *Ladybug* camera from Point Grey Research [2] (Fig.2) contains six closely located and synchronized 1024x768 CCD imagers. [11] and [15] are two examples using this novel sensor.

B. *Cube format*

A panoramic image collects all or part of the light incident on a point in space. People typically think of such a data set as a spherical image, however this does not lend itself to efficient handling. If a six-sided cube format is instead used, virtual perspective images can be more readily handled. The cost of increased storage space (nearly doubled, $\frac{6}{\pi} = 1.9$) over a spherical representation is offset by the benefits of fast rendering with standard graphics hardware [4] and the ease of vision and image processing operations developed for perspective projection images, such as intermediate view generation by image based rendering.

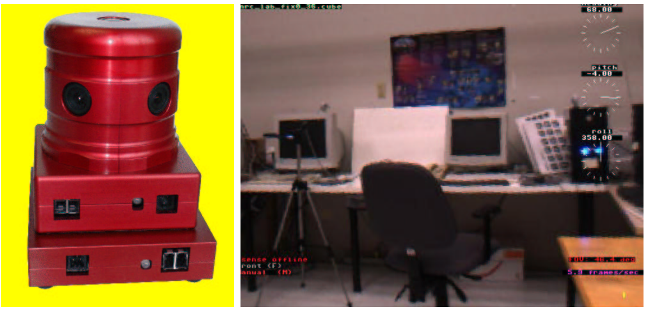

Fig. 2. *(Left)* Ladybug *multi-sensor panoramic camera. (Right) Virtual perspective view created by warping a section of the panoramic image from the Ladybug.*

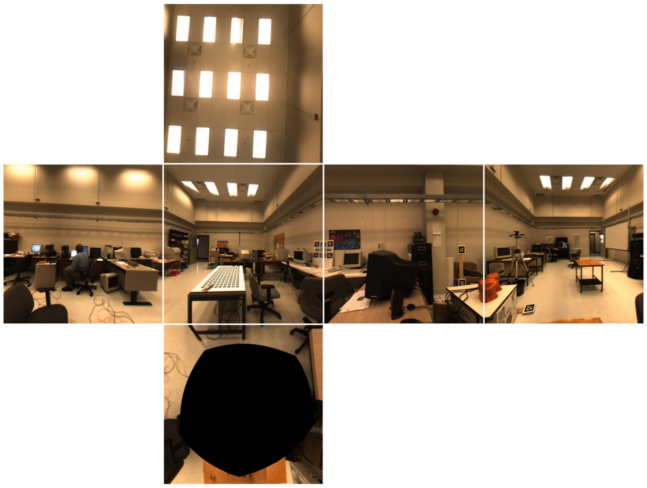

Fig. 3. *Data contained in a ”cube”, six perpendicular perspective images are stored representing all light rays incident to a world point. This cube is one of the ones used in the experiments in this paper, it was captured from the Point Grey Ladybug camera.*

Rendering this cube from a virtual camera located at the cube center that can rotate (but not translate), reproduces all views seen from that point without noticing the cube boundaries. This can be useful for immersively viewing the cube scene with an HMD and orientation tracker as in [7]. An example cube is shown in Fig.3 in a flattened out form. The view seen in the HMD screen is a perspective view that can see up to three cube sides at once, the view is rendered with simple texture mapping.

[1] http://www.cis.upenn.edu/ kostas/omni.html
[2] www.ptgrey.com

C. SIFT Feature Extraction from Panoramic Images

Interest points are used to find correspondences between images, or in our application, panoramas. Interest points are image points that have a strong set of local gradients that distinguish that point. The ideal interest point is a corner in a checkerboard which has a set of orthogonal gradients that identify it's location and orientation. Typically, an interest point operator finds hundreds to thousands of points in a given image in a fraction of a second to a few seconds. The most advanced interest points operators return a sub-pixel location plus an orientation and scale. The local image patch at the given scale and orientation around this interest point is then normalized and from this patch a descriptor vector is computed. This vector should describe the interest point region uniquely, so that similar interest points have a small Euclidian distance between their descriptors.

Currently one of the best known and most successful interest point detectors is the SIFT operator [12] which uses a Difference of Gaussians (DOG) detector to isolate the location, scale and orientation of each interest point. Using the dominant orientation of each interest point, a descriptor is computed from the gradients of the image patch around the interest points at this computed scale and orientation. A normalized histogram of these image gradients is found and a 128 element vector of these gradient orientations is used as the descriptor for this interest point. An image section annotated with SIFT features (square boxes) is shown below in Fig.4 Many SIFT features will find matches in panoramic images taken from similar locations, providing pairs of corresponding points between a pair of panoramic images.

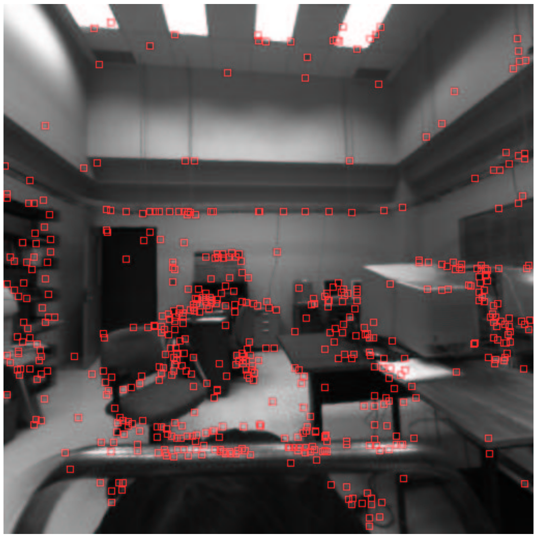

Fig. 4. *SIFT features found in the front face of a cube panorama captured from the Ladybug panoramic camera. A SIFT feature (overlaid as squares) consists of a sub-pixel image location, a scale and orientation, and a 128 byte feature vector describing the neighborhood of the point.*

Correspondences between image frames are accomplished by comparing these keypoint descriptors, so the comparison is done quickly with this abstract data rather than further processing between images (such as correlating image patches to find correspondences). The keypoint descriptors are 128 bytes vectors, the euclidean distance between a SIFT feature in one image, and a potential match in another image is found. A match is declared if the ratio of distances between the closest and second closest keypoint descriptor in the other image [12].

In our experiments, SIFT features were found for each panoramic image by merely processing each cube side separately and amalgamating the SIFT keypoint list. Finding SIFT features in a panorama by simply processing each side image assumes functionality of the SIFT operator up to a 45 degree normal angle, as assumption that worked in our experiments. One could express the panorama in different views to give to the feature detector if this was a problem.

II. ALIGNING PANORAMIC IMAGES

Each cube panorama is captured from a point in space, the cube side images are a representation of the *plenoptic function* for a given world point $[x\ y\ z]^\top$ as a certain orientation. Two different cube panoramas will have different world points and orientations, expressed by a translation vector $\mathbf{t}$ and rotation matrix $\mathbf{R}$. In many applications, only the rotation $\mathbf{R}$ is needed, for example when rectifying cube panoramas for use in stereo matching for model-making and image based rendering of intermediate images. Another application is when only GPS data is available providing position but not reliable orientation. For building a cube map for an interactive walk-through, it is desirable to know this $\mathbf{t}$ and $\mathbf{R}$. However with only two cubes captured with an unknown distance between them, only the direction of translation can be calculated (*i.e.* the translation up to a scale factor).

Herein both cases are considered for a pair of cubes; that of calculating only the rotation matrix, and that of calculating both the rotation matrix and the direction of translation.

In both cases, a set of correspondences are needed. A point in a panorama can be expressed as a direction vector $[x\ y\ z]^\top$. In our experiments, the coordinates $[u\ v]^\top$ within each cube side image is mapped to a vector $[x\ y\ z]^\top$ according to which side it is on. Each cube is processed by finding SIFT features on its 6 sides, producing a list of SIFT features (Fig.4) for the cube. Each SIFT feature has a vector $[x\ y\ z]^\top$ as well as the 128 byte keypoint descriptor. Each SIFT feature list from the first cube is compared to that of the second, by comparing the keypoint descriptors creating a list of corresponding 3D vector pairs (Fig.5). Each vector pair $(x_1, y_1, z_1, x_2, y_2, z_2)$ contains the direction of a feature in the environment as seen in the first and second cube. This list of corresponding vector pairs is created between a pair of cubes, and used to calculate either just the rotation matrix, or both the rotation matrix and the direction of translation.

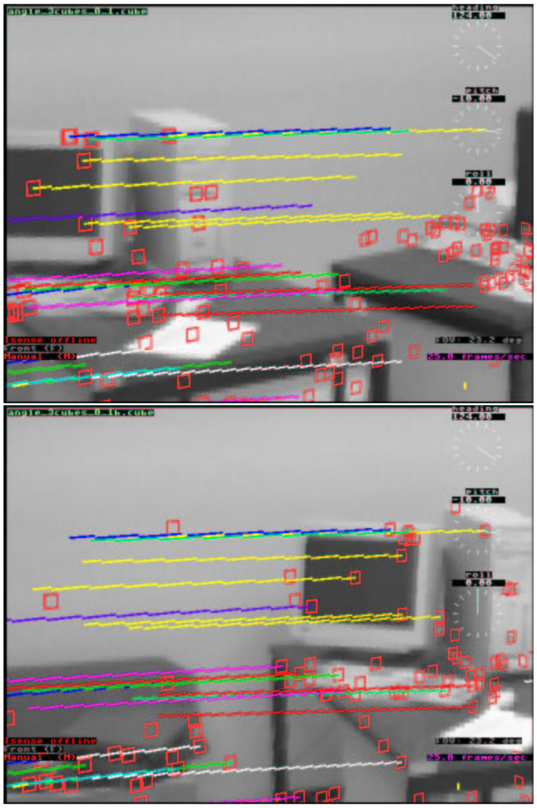

Fig. 5. *Matches of SIFT features between cube panoramas. Section of forward view shown. (Top) SIFT features and matches overlaid over first cube panorama, (bottom) second cube panorama with SIFT features and matches overlaid.*

For the rotation matrix case, the translation is neglected and the difference between two cube panoramas assumed to be due only to rotation. This assumption was found to hold with translations present in our experiments (1 metre translations in a room 16 metres long). Each vector pair $\mathbf{v}_1 = [x_1\ y_1\ z_1]^\top$ and $\mathbf{v}_2 = [x_2\ y_2\ z_2]^\top$ would then satisfy Eqns.1,2. With many matches (such as the 1000+ SIFT vector pairs found per cube pair) a least squares fit can be found for the elements of $\mathbf{R}$.

With the second case, that of finding both rotation and direction of translation, an *essential matrix* $\mathbf{E}$ is instead found. It maps a vector $\mathbf{v}_1$ in the first cube to a plane in the second cube that the corresponding vector $\mathbf{v}_2$ must lie upon. The plane is defined by the normal vector $\mathbf{n}$ ($\mathbf{v}_2$ x $\mathbf{n} = 0$), where $\mathbf{n} = \mathbf{E}\cdot\mathbf{v}_1$. $\mathbf{n}$ can be interpreted as being perpendicular to $\mathbf{v}_1^r = \mathbf{R}\cdot\mathbf{v}_1$ and the translation vector $\mathbf{t}$. The rotation and translation vectors are encompassed in the standard essential matrix by a cross product operation [3].

Thus an essential matrix $\mathbf{E}$ is found that satisfies Eqns. 3,4 from all the vector pairs between two cube panoramas, this can be solved using least squares and SVD methods, and the rotation $\mathbf{R}$ matrix and translation $\mathbf{t}$ vector extracted from $\mathbf{E}$ as per existing methods [4]. This is a more general case of the essential matrix found as part of a *fundamental matrix* which relates points in one standard (non-panoramic) perspective view with lines in another view. With panoramas in the cube format, there is no camera calibration matrix and the direction vector can simply be taken from the position of a pixel on the cube.

$$\mathbf{v}_2 = \mathbf{R}\mathbf{v}_1 \tag{1}$$

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{2}$$

$$\mathbf{v}_2^\top \mathbf{E}\mathbf{v}_1 = 0 \tag{3}$$

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{4}$$

A. *Experiments*

Two experiments were performed with the first case of cube alignment: calculating the rotation matrix only. The first experiment was with three cube panoramas taken at the same location with varying azimuth (heading) angles, a rotation matrix $\mathbf{R}$ was found between cubes 1 & 2, and between 2 & 3. SIFT features were found, matched creating lists of corresponding vector pairs, each list was converted to a rotation matrix as per Eqn. 2. The azimuth between cubes 2 & 3 was added to that found between cubes 1 & 2, and cubes 2 & 3 were thus aligned to face the same direction as cube 1. Forward views of the original and aligned cubes are shown in Fig.6.

A set of 19 images were taken with substantial rotation and translation, the panoramic camera was moved around a path detailed in Section III below. The path stretched over 7 metres by 2.5 metres in a room of dimensions about 8 x 16 metres, despite this relative large translation with respect to the environment (and thus the SIFT features) the resulting image sequence was able to be aligned. The error between the first and last cube was about 10°, the error is propagated since only the relative angle between each cube pair can be found. The relative angle error between cube panoramas was smaller.

III. IMAGE SEARCH TO FIND SELF-INTERSECTIONS (CROSS-OVER POINTS)

A cube panorama is the re-projection of a panoramic image onto a six sided cube. Assume the cubes are sampled by moving the panoramic camera along a path, and that this path contains self-intersections. For example, the panoramic camera may be moved along a figure eight pattern, and the path crosses itself at the center of the eight. The goal is to find these types of intersections automatically. Assume that there are n

[3] A 3x3 matrix containing elements of $\mathbf{t}$ arranged so a multiplication is the same as a cross product, is pre-multiplied by the 3x3 $\mathbf{R}$ to provide the 3x3 essential matrix $\mathbf{E}$

[4] Section 5.3 of [13], or Chapter 5 of [5]

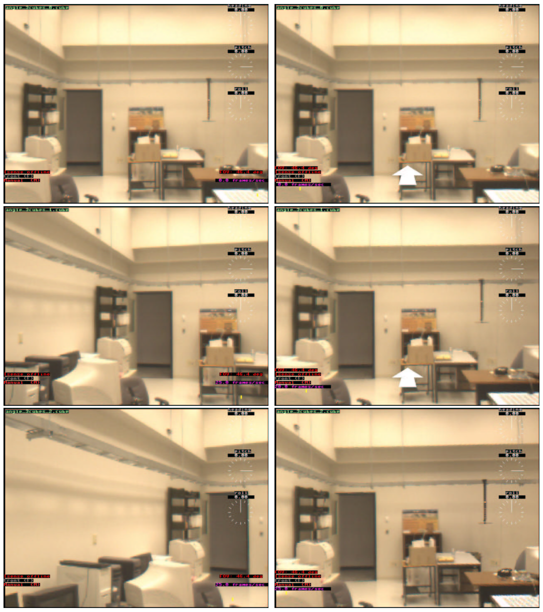

Fig. 6. *Automatic rotational alignment of cubes. (Left column) front view from three cube panoramas captured about $12°$ apart. (Right column) front view of corresponding cubes aligned automatically in azumith.*

such cube panoramas, and that each of these images contains the six sides of the cube. This means that each cube panorama must be matched against every other cube panorama, which is an $O(n^2)$ process. For this reason such an exhaustive matching procedure is currently practical only for n smaller than fifty. However, in the future more sophisticated indexing methods may make it possible to find the self intersections for very long sequences of cube panoramas.

The result of a single match is the number of SIFT features between the two cubes. Note that all six faces of the cube panoramas are matched at once. The ideal result is that the closest match to any cube panorama at an intersection point is the image that represents the next camera position, or the image that is the result of the intersected path.

An experiment was performed where the panoramic Ladybug camera was moved onto a path (Figure 7) defined by a double figure eight. Each of the cubes are numbered from zero to eighteen, consecutively along the path of the capture. It is clear from the figure that there are two intersection points, the group $4, 5, 15, 16$, and the group $7, 8, 12, 13$, as well as a point of close proximity at $1, 2, 18$. A list of SIFT features is extracted from every cube and used to exhaustively compare to all the other cubes' SIFT feature lists. In our experiment, adjacency is simply determined by the best number of SIFT matches. For each cube, an ordered list of the other cubes is made according to how many matches were found. Ideally the first few in this list should be the preceding and following cubes in the path, and cubes from another part of the path with near proximity (such as a cross-over).

We perform an exhaustive match of each cube against all other eighteen cubes, and then we find the top two matches for each cube. For the cubes not near a self-intersection, the nearest matches are the preceding and following cube as expected. In Table III we show the first two cubes in the ordered list of matching cubes (only the potential intersections are shown). The table shows that the top two cube matches are indeed correct; one is the next cube in the sequence, and the other is the cube that is on the self-intersection. This experiment demonstrates that this exhaustive matching process is capable of finding the self-intersections.

Query Image	First Match	Second Match
1	18	2
2	18	1
4	15	5
5	16	4
15	4	16
16	15	5
7	12	8
8	7	13
12	7	13
13	12	8

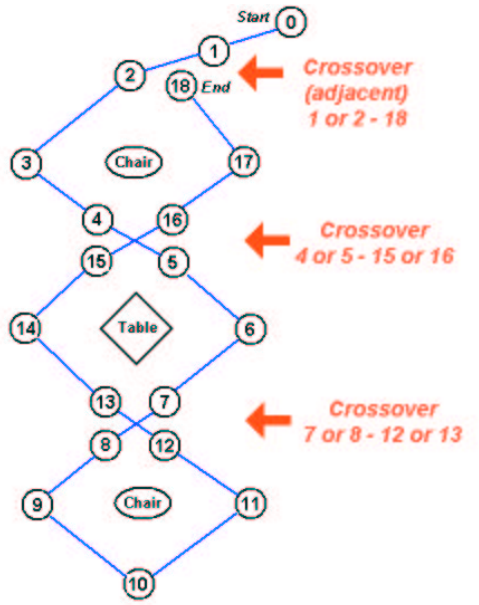

Fig. 7. *Matches of SIFT features between cube panoramas. Section of forward view shown. (Top) SIFT features in first image, (middle) overlay of matches with SIFT features in second image, (bottom) second image with SIFT features and matches overlaid.*

IV. CONCLUSIONS

A panoramic video camera provides a linear stream or sequence of panoramas, it is desirable to calculate the inter-panorama alignment and a global map without relying on other sensors. Panoramic images were stored in cube form, and SIFT features found on the cube sides to find correspondences between cubes. This paper presented two methods to align two cubes, one finding only rotation by finding a rotation matrix, and one finding both rotation and translation direction by finding an essential matrix. Also, an image search based approach was used to find cross-over (self-intersections) in the camera path.

[1] http://www.site.uottawa.ca/research/viva/projects/ibr.

[2] S. Baker and S. Nayar. A theory of catadioptric image formation. In *IEEE ICCV Conference*, 1998.

[3] A. Basu and D. Southwell. Omni-directional sensors for pipe inspection. In *IEEE SMC Conference*, pages 3107–3112, Vancouver, Canada, October 1995.

[4] D. Bradley, A. Brunton, M. Fiala, and G. Roth. Image-based navigation in real environments using panoramas. In *HAVE 2005: IEEE International Workshop on Haptic Audio Visual Environments and their Applications*.

[5] O. Faugeras and Q. Luong. *The Geometry of Multiple Images*. MIT Press, Cambridge, Massachusetts, 2001.

[6] M. Fiala. Pano-presence for teleoperation. In *Proc. of IROS 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (in print)*.

[7] M. Fiala. Immersive panoramic imagery. In *Proc. of CRV'05 (Canadian Confernence on Computer and Robot Vision*, pages 386–391, May 2005.

[8] M. Fiala. Structure from motion using sift features and the ph transform with panoramic imagery. In *Proc. of CRV'05 (Canadian Confernence on Computer and Robot Vision*, pages 506–513, May 2005.

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

[10] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. Cambridge, UK, 2000. Cambridge University Press.

[11] S. Ikeda, T. Sato, and N. Yokoya. High-resolution panoramic movie generation from video streams acquired by an omnidirectional multi-camera system. In *Proc. IEEE Intl. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, volume 2, pages 155–160, 2003.

[12] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60:91–110, 2004.

[13] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An Invitation to 3-D Vision*. Springer-Verlag Inc, New York, New York, 2004.

[14] T. Sato, S. Ikeda, and N. Yokoya. Extrinsic camera parameter recovery from multiple image sequences captured by an omni-directional multi-camera system. In *Proc. European Conf. on Computer Vision*, volume 2, pages 326–340, 2004.

[15] M. Uyttendaele, A. Criminisi, S. Kang, S. Winder, R. Hartley, and R. Szeliski. High-quality image-based interactive exploration of real-world environments. In *Microsoft Research Technical Report*, October 2003.