Space Variant Feature Extraction for Omni-directional Images

Dermot Kerr[1], Bryan Scotney[2], Sonya Coleman[1]

[1]School of Computing and Intelligent Systems
University of Ulster
Magee, BT48 7JL
{kerr-d2, sa.coleman}@ulster.ac.uk

[2]School of Computing and Information Engineering
University of Ulster
Coleraine, BT52 1SA
bw.scotney@ulster.ac.uk

Abstract

In recent years, the use of omni-directional cameras has become increasingly more popular in vision systems and robotics. To date, most of the research relating to omni-directional cameras has focussed on the design of the camera or the way in which to project the omni-directional image to a panoramic view rather than on how to process these images after capture.

Typically images obtained from omni-directional cameras are transformed to sparse panoramic images that are interpolated to obtain a complete panoramic view prior to low level image processing. This interpolation presents a significant computational overhead with respect to real-time vision. We present an approach to real-time vision that projects an omni-directional image to a sparse panoramic image and directly processes this sparse image. Feature extraction operators previously designed by the authors are used in this approach but this paper highlights the reduction of the computational overheads of processing images arising from omni-directional cameras through efficient coding and storage, whilst retaining accuracy sufficient for application to real-time robot vision.

Keywords: Omni-directional imaging, Feature detection.

1 Introduction

Panoramic vision systems have been developed based on the natural biological vision systems of arthropods such as insects [1]. The arthropod's vision system is simple when compared with the complex eye movement system of humans. Arthropods have compound eyes that, although immobile with respect to their body, have a natural advantage gained from a wide field of view [2]. This wide field of view would be of benefit to mobile robot localisation, as it allows many landmarks to be simultaneously present in the scene, without the need for an elaborate pan-tilt active vision system that would mimic human gaze control. Catadioptric cameras combine conventional cameras and mirrors to obtain a 360° omni-directional view similar to the wide field of view that a compound eye obtains. The camera is pointed vertically towards a mirror, with the optical axis of the camera lens aligned with the mirror's axis. The mirror profiles that can be used are conical or convex, such as spherical, parabolic, and hyperbolic. These cameras offer advantages to robot navigation systems mainly due to the fact that they provide the robot with a complete 360° omni-directional view of its immediate surroundings [3, 4], allowing many landmarks to be in view of the robot simultaneously, and they offer the ability to the robot to change viewing direction instantaneously. They enable a single feature to be tracked from varying viewpoints, whereas if a fixed camera were used, the feature may track out of the field of view. However, their disadvantage is that they capture images of a lower resolution than a standard image. The reduced resolution may be a problem for tasks such as fine manipulation or control, but for navigation tasks the wide field of view would be of greater value than higher resolution.

Previous research using omni-directional images for robot navigation includes Gaspar [1], in which the omni-directional image is un-warped to both panoramic and birds-eye-view images. An appearance based navigation system is used that has a set of reference birds-eye-view images and a topological map of the environment for navigation. Another appearance based method is used by Matsumoto [5] in which omni-directional images are un-warped to a panoramic view before template matching of images is performed. Yagi [6] also un-warps the omni-directional image to a

panoramic view before detecting edges using the Sobel edge detector. The edge maps are then used to create a one dimensional projections relating to the edge density of the horizontal axis of the edge map for the purpose of navigation. A different approach is taken by Vlassis [3], where the omni-directional image is not un-warped. In this approach the Sobel edge detector is used to extract edges from the omni-directional image, and the edge pixels are then fed to a Parzen density estimator to calculate the edge density. Principal component analysis is then used to reduce the dimensionality of the feature vector that is used in the navigation system.

As discussed above, the omni-directional image may be used in the original format [3], or it may be un-warped to a rectangular panoramic image through the use of an un-warping algorithm [7, 8, 9]. In order to un-warp the omni-directional image to a panoramic image, it is typical for the circular omni-directional image to be projected onto a cylinder, transforming the omni-directional image to an un-warped rectangular panoramic image. Previous research has highlighted advantages of un-warping the omni-directional image. Matsumoto [5] highlights the fact that a system that can deal with shifts of panoramic images using template matching in hardware has lower computational and storage costs than a system that can cope with rotations of omni-directional images. Krose [10] also cites this as a benefit, along with the fact that standard feature trackers may be used on the panoramic image.

In this paper we discuss our method of extracting features directly from a sparse un-warped panoramic image. We will show that our method retains the accuracy of traditional 4- and 8-pixel neighbourhood feature extraction operators on complete images, whilst achieving a reduced computational overhead due to avoidance of image interpolation. Section 2 discusses methods of un-warping an omni-directional image to a panoramic image. Section 3 introduces our method of space variant feature detection, while Section 4 discusses implementation and implementation issues such as efficient coding and storage. In Section 5 we present processing times and sample edge maps using various edge detection methods for comparison, and Section 6 summarises our findings and outlines future work.

2 Sparse Panoramic Images

We obtained omni-directional images, as shown in Figure 1, using a mobile robot with a catadioptric camera, shown in Figure 2.

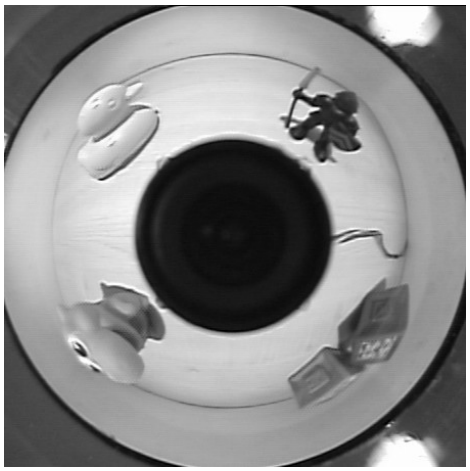


Figure 1. Omni-directional image

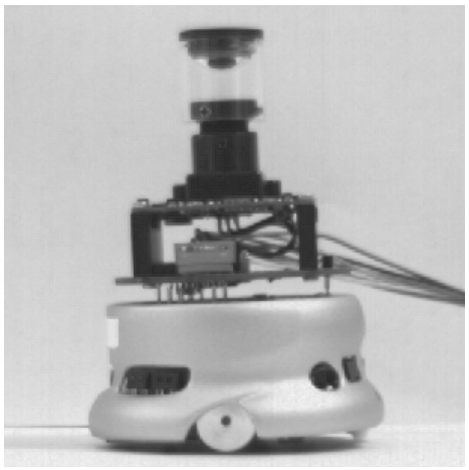


Figure 2. Khepera robot used to obtain images

One approach to un-warping of omni-directional images is by back-projection of each pixel in an un-warped panoramic image to a position in the omni-directional image by means of a polar-to-cartesian transformation:

$$g(r,\theta) = f(x,y) = f(r\cos\theta, r\sin\theta) \tag{1}$$

where $g(r,\theta)$ represents the polar coordinate image and $f(x,y)$ represents the rectangular coordinate image. The value $g(r,\theta)$ in the omni-directional image is obtained by bilinear interpolation of the four pixel values in the 4-pixel neighbourhood of (r,θ). A three-quarter section of the un-warped version of the omni-directional image shown in Figure 1 obtained by

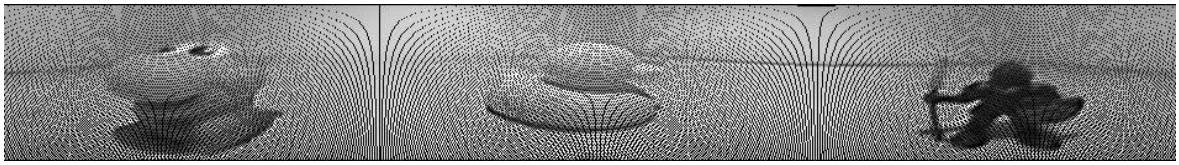
using back-projection method is shown in Figure 3(a). Although bilinear interpolation is generally cheap to implement, such a back-projection has the disadvantage of relying on secondary reconstructed image data.

A second approach to un-warping is to forward-project each pixel in the omni-directional image to a position in the un-warped panoramic image by means of a cartesian-to-polar transformation:

$$f(x,y) = g(r,\theta) = g\left(\sqrt{x^2+y^2}, \tan^{-1}\frac{y}{x}\right) \tag{2}$$

where $f(x,y)$ represents the rectangular coordinates image and $g(r,\theta)$ represents the polar coordinates image. This un-warping approach leads to a sparse representation of the panoramic view, as shown in Figure 3(b), where the missing values depicted in black. Typically post-processing would be required, such as applying interpolation to this image, in order to obtain a complete data set before any further image processing. Our approach however is to work directly on the space variant panoramic image, thus avoiding the use of secondary reconstructed image data values.

(a)



(b)

Figure 3. (a) Back-projected un-warped image; (b) Forward-projected un-warped image

3 Space Variant Operators

In order to extract features from the space variant panoramic image described in Section 2 in real-time, we use the family of autonomous finite element based image processing operators presented in [11] for use on non-uniformly sampled intensity images and in [12] for use on range images. Here, the term autonomous indicates that these operators were developed in such a way that they can change size and shape across the image plane in accordance with the local pixel distribution as illustrated in Figure 4.

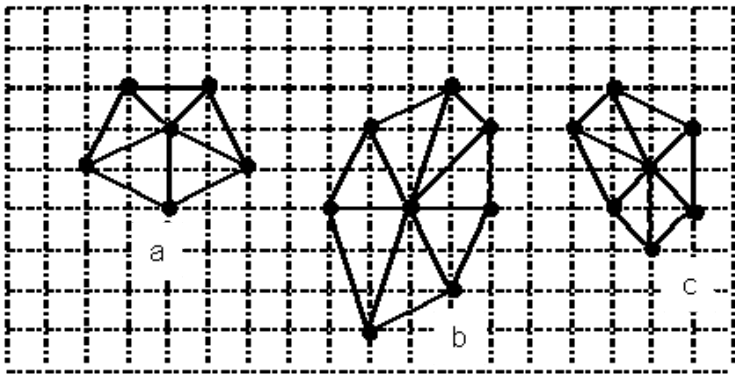


Figure 4. Neighbourhoods of Autonomous Operators

These feature detection operators correspond to weak forms of operators in the finite element method and can be based on first or second order derivative approximations, corresponding to a first directional derivative $\partial u/\partial b \equiv \underline{b}\cdot\underline{\nabla}u$ and a second directional derivative $-\underline{\nabla}\cdot(\mathbf{B}\underline{\nabla}u)$, and are defined by the functionals

$$E_i^{\sigma}(U) = \int_{\Omega} \underline{b}_i \cdot \underline{\nabla} U \psi_i^{\sigma} \, d\Omega \quad \text{and} \quad Z_i^{\sigma}(U) = \int_{\Omega} \underline{\nabla} U \cdot (\mathbf{B}_i \underline{\nabla} \psi_i^{\sigma}) d\Omega \,. \tag{3}$$

Here, U is our representation of the image data, $\underline{b} = (\cos\theta, \sin\theta)$ is an image-dependent unit direction, $\psi_i^{\sigma_m}(x,y) = \dfrac{1}{2\pi\sigma_m^{\,2}} e^{-\left(\frac{(x-x_i)^2+(y-y_i)^2}{2\sigma_m^{\,2}}\right)}$ and $\mathbf{B} = \underline{b}\,\underline{b}^{\mathrm{T}}$. In the special case of the Laplacian operator, represented by Z_i^{σ}, $\mathbf{B}$ is taken to be the identity matrix. By choosing each function ψ_i^{σ} to be a Gaussian function restricted to a neighbourhood Ω_i^{σ}, each operator is restricted to the neighbourhood Ω_i^{σ}.

As these operators can be applied directly to space variant image data, we can apply them directly to the forward-projected un-warped panoramic image without the need for additional image reconstruction. The application of these operators to space variant panoramic images yields edge maps that are comparable to those obtained using well-known image processing operators on complete panoramic images, as will be illustrated in Section 5.

4 Implementation

As shown in Figure 2, our system uses a Khepera II miniature robot manufactured by K-Team Corporation of Switzerland. The Khepera II has the optional K2D camera turret, which utilises a camera pointed upwards, looking at a spherical mirror to acquire omni-directional images. 3-channel colour omni-directional images are captured using a Matrox Meteor II frame-grabber at full resolution of 510×492. Images are stored, processed, and displayed on our graphics workstation using the *Open Source Computer Vision Library (OpenCV)* [13]. The 3-channel omni-directional images were then converted to 1-channel grey-scale images using the `ConvertToGrey()` function in *OpenCV*, before being centred and cropped to a final resolution of 452×452. An example of a cropped omni-directional image is shown in Figure 1.

The image is un-warped using the forward-projection method, as described in Section 2, to create a space variant panoramic image. The space variant image data are used in Delauney triangulation to construct a triangular mesh, defined by a set $T = \{e_m\}$ of triangular elements, on which our feature detection operators are constructed. To create this mesh we have incorporated the '*Triangle*' meshing library [14] into our application. Information on nodal connectivity is stored by creating a list of connected nodes for each node in the image; the compactness of the list structure can be improved by judicious global ordering of the mesh nodes.

With each node (x_i, y_i) is associated an image value U_i and a piecewise linear basis function $\phi_i(x,y)$ spanning a neighbourhood Ω_i^{σ} comprised of the set S_i^{σ} of triangular elements that share (x_i, y_i) as a vertex. The approximate image representation may then be written as

$$U(x,y) = \sum_{j=1}^{N} U_j \phi_j(x,y) \tag{4}$$

and is therefore piecewise linear on each triangle, and this image representation is used in equation (3) to develop feature detection operators that correspond to local derivative approximations.

In order to compute the weights in an operator we need to compute *element integrals* $k_{ij}^{m,\sigma}$, each defined over an element e_m in the neighbourhood set S_i^{σ}. For example, a contribution to a gradient operator $E_i^{\sigma}(U)$ would be $k_{ij}^{m,\sigma} = \int_{e_m} \dfrac{\partial \phi_j}{\partial x} \psi_i^{\sigma} \, dxdy$, whilst a contribution to a Laplacian operator $Z_i^{\sigma}(U)$ would be $k_{ij}^{m,\sigma} = \int_{e_m} \dfrac{\partial \phi_j}{\partial x} \dfrac{\partial \psi_i^{\sigma}}{\partial x} dxdy$. Such element integrals are accurately and efficiently evaluated using low-order (3-point) Gaussian quadrature. In finite element analysis, the $k_{ij}^{m,\sigma}$ values are stored in small *finite element matrices*: for a triangular element, each finite element matrix is 3×3 in size, with each row corresponding to a particular node in the triangle. A complete set of space variant autonomous operators, $E_i^{\sigma}(U)$ or $Z_i^{\sigma}(U)$, may thus be computed by

assembling the $k_{ij}^{m,\sigma}$ values for each element e_m in S_i^σ according to information on nodal connectivity routinely stored in look-up-tables; i.e., each operator weight may be computed as $K_{ij}^\sigma = \sum_{\{m|e_m \in S_i^\sigma\}} k_{ij}^{m,\sigma}$.

The amount of computation required to carry out image un-warping by forward projection followed by construction of the triangular mesh and then the corresponding feature detection operators is greater than that required to construct a complete un-warped image by backward projection followed by application of traditional 4- or 8-pixel low level processing techniques. Hence, it might seem that real-time application is precluded. However, this is not the case, as the cartesian-to-polar transformation required for forward projection needs to be computed only once. Each pixel in an omni-directional image transforms to a fixed position in the un-warped panoramic image, and in all subsequent un-warped panoramic images. Hence for the robot to process a whole sequence of omni-directional images, we need to construct the triangular mesh and operators only once. By storing the operators in an efficient look-up-table, computation becomes sufficiently reduced to enable real-time application.

Whilst the operators could be constructed by computing and storing the finite element matrices described above, our performance experiments show that it can be more efficient to use a combination of stored information and some on-the-fly-computation; the best balance is achieved by minimizing the number of floating-point values that need to be addressed in look-up-tables. Since the image representation in equation (4) is piecewise linear on each element, the derivative terms, $\frac{\partial \phi_j}{\partial x}$ and $\frac{\partial \phi_j}{\partial y}$, in the element integrals $k_{ij}^{m,\sigma}$ are constant within an element, depending only on the locations of the three nodes of the element. Hence, rather than store and retrieve three fully computed element integrals $k_{ij}^{m,\sigma} = \int_{e_m} \frac{\partial \phi_j}{\partial x} \psi_i^\sigma \, dxdy$ or $k_{ij}^{m,\sigma} = \int_{e_m} \frac{\partial \phi_j}{\partial y} \psi_i^\sigma \, dxdy$ as floating point values on each element for each operator, it is more efficient to store and retrieve the single floating point value $\int_{e_m} \psi_i^\sigma \, dxdy$ for each element and multiply this value on-the-fly by the three constant values corresponding to $\frac{\partial \phi_j}{\partial x}$ or $\frac{\partial \phi_j}{\partial y}$. Figure 5 shows a graphical representation of how the operator data is stored in the look-up-table

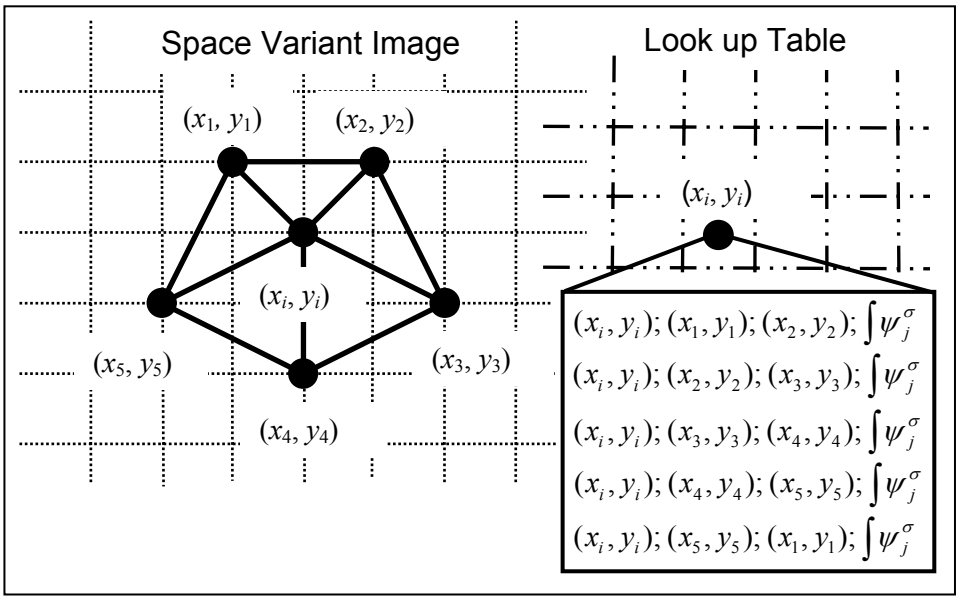


Figure 5. Graphical representation of operator data storage in look-up-table

The procedure for the initial creation of the operators and subsequent processing of images is summarised in the following pseudo-code:

```
Capture first omni-directional image
Convert to grey-scale and crop
Un-warp to sparse panoramic view
Apply triangulation using 'Triangle'
For i=1 to numNodes
/* for each node in the sparse image*/
     for j=1 to numElements
      /* for each element in a local neighbourhood */
      compute scale parameter σ for element
      compute and store ψ_j^σ integral in look-up-table
      store element node co-ordinates
      }
}

/* for all subsequent omni-directional images */
Convert to grey-scale and crop
Un-warp to sparse panoramic view
For i=1 to numNodes
/* for each node in the sparse image */
     for j=1 to numElements
      /* for each element in look-up-table */
      compute X = Σ∫ (∂φ_i/∂x) ψ_j^σ dxdy × U_i
      compute Y = Σ∫ (∂φ_i/∂y) ψ_j^σ dxdy × U_i
      operator response= √(X^2 + Y^2)
      }
}
```

In addition to the adaptive method described above, for purposes of comparison we have also implemented the back-projection technique described in Section 2, producing a complete image that may be used with standard feature detection techniques. We have applied the Sobel, Prewitt, and Canny edge detection operators to this full image for comparison with our results presented in Section 5.

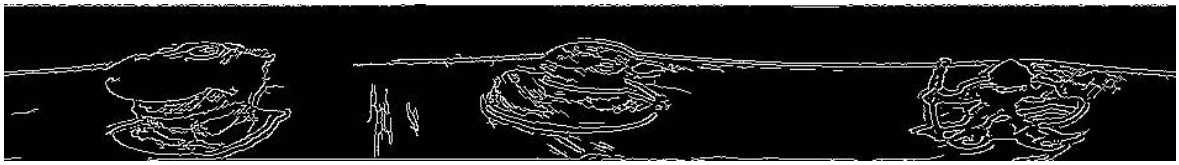
5 Experimental Results

We present processing times that consist of un-warping time and time taken to extract edges from the un-warped images. The space variant operator uses sparse images generated by the forward-projection un-warping technique described in Section 2. The Canny, Sobel, and Prewitt operators are used on complete un-warped images obtained by the back-projection technique also described in Section 2. Each algorithm has been run 100 times on a graphics workstation, consisting of a Pentium 4, 3.4 GHz processor and 2GB RAM, and the average processing time calculated. The space variant operator is comparable in processing time to both the Sobel and Prewitt operators, and is faster than the Canny operator. Comparative results for processing are presented in Table 1. These processing times do not include file I/O or capture times. With the space variant method it is also possible to reduce the quantity of data that are used for detecting features. In our experiments we have un-warped the omni-directional image to an extra-sparse panoramic image by using both random and regular sampling of the original omni-directional image data. This has the effect of reducing the quantity of data that needs to be processed, whilst still maintaining the ability to detect features using our space variant technique. Average processing times incorporating random and regular sampling have also been included in Table 1.

Operator	Processing Time (Seconds)
Space Variant (70% data)	0.069309
Space Variant (35% data) random sample	0.051391
Space Variant (35% data) regular sample	0.056221
Prewitt	0.068846
Sobel	0.068541
Canny	0.112076

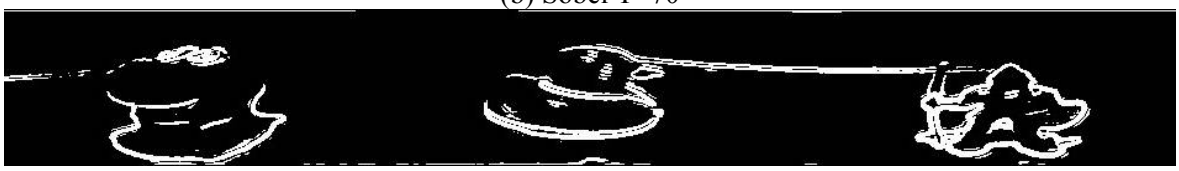
Table 1. Processing times for detecting edges

In Figure 6 we present edge maps obtained using various feature detection techniques on the panoramic image shown in Figure 3. We have selected the visually best edge maps obtained with each feature detector at threshold T, and in the case of the Canny feature detector, threshold low TL, and threshold high TH. Figures 6(d), 6(e), and 6(f) illustrate that our approach yields feature maps containing sufficient information to recognise the objects in Figure 3. Our system captures images at a frequency of 25 Hz and using the space variant approach with a random sample we can forward un-warp and perform edge detection at a frequency of ~20 Hz, while reverse un-warping and using the Canny edge detection method a frequency of ~9 Hz is obtained. These results indicate that real-time performance approaching 20 Hz is achievable using the space variant approach, and also indicate that this approach would be suitable for use in a real-time robot localisation system as no significant delay between capture and processing is present.



(a) Canny TL=20 TH=70

(b) Sobel T=70



(c) Prewitt T=20

(d) Adaptive 70% data T=20

(e) Adaptive random sample 35% data T=20

(f) Adaptive regular sample 35% data T=20

Figure 6. Sample image sections and associated edge maps

6 Summary and Further Work

We have shown that through the use of appropriate data structures a family of autonomous finite element based image processing operators can process un-warped sparse omni-directional images in real-time, producing feature maps of a quality suitable for object recognition. Such an approach has the facility to reduce the quantity of input data further, thereby further reducing the processing times to become significantly faster than the traditional methods (approximately 30% reduction in processing time), whilst still producing edge maps suitable for object recognition. A slight computational time penalty is unavoidable with the initial construction of the mesh, however this is minimised as this mesh has to only be constructed on an initial image and not for subsequent images. The accuracy of the operator is also reliant on the accuracy of the underlying triangulation process. We have previously shown in [12] that the autonomous finite element based image processing operators can successfully process sparse range images. The approach of using autonomous finite element based image processing operators may also be used on other images with a non-uniform resolution such as log-polar images [15]. The next stage in our work is to develop a localisation system for the mobile robot using the feature maps of the robot's environment obtained by the current vision system.

7 Acknowledgements

This work was funded by the Nuffield Foundation under the research grant number NAL/00839/G.

8 References

[1] Gaspar, J., Winters, N., Santos-Victor, J., "Vision-based Navigation and Environmental Representation with an Omni-directional Camera" *IEEE T-RA, Vol. 16, No.6, pp.890-898, 2000*.

[2] Argyros, A.A., Tsakiris, D.P. & Groyer, C., "Biomimetic centering behavior", *IEEE Robotics & Automation Magazine., Vol. 11, No 4, pp. 21-30, 2004*.

[3] Vlassis, N., Motomura, Y., Hara, I., Asoh, H. & Matsui, T., "Edge-based features from omnidirectional images for robot localization", *Proc. IEEE ICRA, pp. 1579-1584, 2001*.

[4] Winters, N., Gaspar, J., Lacey, G., Santos-Victor, J., "Omni-directional Vision for Robot Navigation" *IEEE Workshop on Omnidirectional Vision, p.21, 2000*.

[5] Matsumoto, Y., Ikeda, Y., Inaba, M., Inoue, H. "Visual navigation using omnidirectional view sequence". *Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp. 317--322, 1999.*

[6] Yagi, Y., Hamada, H., Benson, N., Yachida, M. "Generation of stationary environmental map under unknown robot motion". *Proc. IEEE/RSJ IROS, Vol. 2, pp. 1487-1492, Japan, 2000.*

[7] Torii, A. & Imiya, A. 2004, "Panoramic image transform of omnidirectional images using discrete geometry techniques", *3DPVT'04., pp. 608-615, 2004.*

[8] Nayar S.K., "Omnidirectional Vision" *Proc. 8th Int. Symp. Robotics Research, Japan 1997*

[9] Peri, V. & Nayar , S., "Generation of perspective and panoramic video from omnidirectional video*", In Proc. of 1997 DARPA Image Understanding Workshop, New Orleans., 1997.*

[10] Krose, B., Bunschoten, R., Hagen ST, Terwijn, B. & Vlassis, N., "Household robots look and learn: environment modeling and localization from an omnidirectional vision system", *IEEE Robotics & Automation Magazine, vol. 11, no. 4, pp. 45-52, 2004*.

[11] Coleman, S.A., Scotney, B.W. "Autonomous Operators for Direct use on Irregular Image Data" *Proceeding of the 2005 ICIAP, pp. 296-303, LNCS 3617, Springer Verlag.*

[12] Coleman S.A., Scotney B.W., Kerr D., "Direct Feature Extraction on Range Image Data", *Proc. of Irish Machine Vision and Image Processing Conference, QUB, Belfast, 2005.*

[13] Bradski, G.R., Pisarevsky, V. "Intel's Computer Vision Library: applications in calibration, stereo segmentation, tracking, gesture, face and object recognition", *IEEE CVPR, Vol. 2, pp. 796-797, 2000*.

[14] Shewchuk, J., "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator", *Applied Computational Geometry', Springer-Verlag, Berlin, Vol. 1148, pp. 203-222, 1996.*

[15] Wallace, R., Ong, P., Bederson, B., Schwartz, E., "Space variant image processing", *International Journal of Computer Vision, Vol. 13, No. 1, pp. 71-90, 1994.*