Ego-Motion and Omnidirectional Cameras*

Joshua Gluckman and Shree K. Nayar
Department of Computer Science
Columbia University
New York, New York 10027

Abstract

Recent research in image sensors has produced cameras with very large fields of view. An area of computer vision research which will benefit from this technology is the computation of camera motion (ego-motion) from a sequence of images. Traditional cameras suffer from the problem that the direction of translation may lie outside of the field of view, making the computation of camera motion sensitive to noise. In this paper, we present a method for the recovery of ego-motion using omnidirectional cameras. Noting the relationship between spherical projection and wide-angle imaging devices, we propose mapping the image velocity vectors to a sphere, using the Jacobian of the transformation between the projection model of the camera and spherical projection. Once the velocity vectors are mapped to a sphere, we show how existing ego-motion algorithms can be applied and present some experimental results. These results demonstrate the ability to compute ego-motion with omnidirectional cameras.

1 Introduction

Recently, researchers have proposed and implemented wide-angle image sensors capable of imaging panoramic, hemispherical, and spherical fields of view [11] [12] [17] [24]. Autonomous navigation, remote surveillance, and video conferencing are among the applications which should benefit from this technology. Already, wide-angle imaging devices have begun to be incorporated into autonomous navigation systems [3] [15] [22]. In this paper, we discuss the use of omnidirectional cameras for the recovery of observer motion (ego-motion), an important problem in autonomous navigation.

The ego-motion problem can be stated as the recovery of observer rotation and direction of translation at a given instant of time, as the observer moves through the environment. Although, in principle, information about observer motion is present in the motion field, in practice it has proven to be difficult to extract. Vision researchers have developed a multitude of algorithms for solving the ego-motion problem. Most methods comprise two steps. The first is motion-field estimation, or the computation of optic flow. The second is motion field analysis, which is the extraction of camera translation and rotation from the optic flow. One of the problems associated with the computation of ego-motion is the sensitivity of the second step to noisy estimates of optic flow. By using omnidirectional camera systems, we seek to overcome this problem.

It is well known that a large field of view facilitates the computation of observer motion. The motion field contains global patterns which do not always manifest

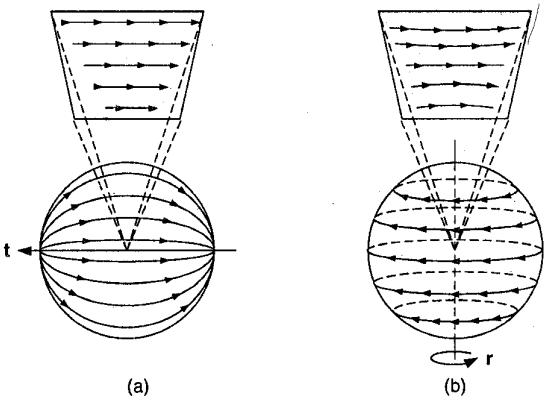

Figure 1: (a) *The projection of a translational motion field onto a sphere and onto a plane parallel to the direction of translation* **t**. (b) *The projection of a rotational motion field onto a sphere and onto a plane parallel to the rotation vector* **r**. *Note that the planar flow fields are nearly the same for the two different types of motion, whereas the spherical flow fields are considerably different. In fact, over half the sphere the flow is in the opposite directions. This simple example illustrates the advantage of having a large field of view.*

*This research was supported in parts by the DARPA Image Understanding Program on Video Surveillance and Monitoring and an NSF National Young Investigator Award.

themselves in a small field of view. In particular, the focus of expansion is likely to exist outside the field of view of traditional cameras, causing ego-motion algorithms to be sensitive to the orientation of the camera. In contrast, for wide-angle imaging systems with a hemispherical field of view, either the focus of expansion or the focus of contraction will always exist in the image. For a spherical field of view, both the focus of expansion and contraction will exist in the image. Another problem associated with the small field of view of traditional cameras is displayed in figure 1. Translation parallel to the image and rotation about the vertical axis produce similar motion fields when the field of view is small. In the presence of noise, these fields are difficult to disambiguate. However, for a very large field of view (180 degrees or greater) the motion fields are distinct even in the presence of large amounts of noise.

Previously, Yagi *et. al.* used a hyperbolic omnidirectional camera to compute camera motion under the assumption that the camera moves in a horizontal plane [22]. In this paper, we present a general framework for computing ego-motion from image sensors, using no assumptions about the camera geometry or the type of camera movement. By noting that the motion field equations for different camera models are related to one another through a transformation, we propose computing optic flow in the image and then mapping the flow to a spherical projection model using the Jacobian of the transformation. Alternatively, we could project the images onto a sphere and then compute optic flow. However, projecting the images introduces artifacts while projecting the flow field does not. We use the spherical perspective projection model because it is convenient for representing fields of view greater than 180 degrees. In addition, existing ego-motion algorithms can be adapted to spherical projection.

2 Omnidirectional Image Sensors

Traditional camera systems have a small field of view, typically a cone of about 45°. In the literature, there have been several methods proposed for increasing the field of view of traditional camera systems. See [11] and [12] for a more in-depth review of existing techniques.

As noted in [11] [13] and [24], a single center of projection is a desirable property for an imaging system to have. A single center implies that all principle lights rays imaged, pass through a single point in 3-D space. It is this property that allows the generation of pure perspective images and the projection of the image velocity vectors onto a sphere [16].

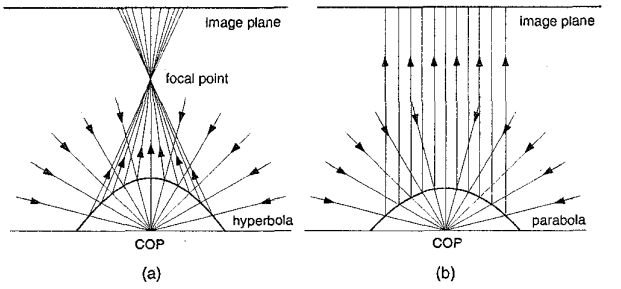

Figure 2: (a) *The geometry of a hyperbola is such that all incoming rays pointed at the interior focal point will be reflected through the exterior focal point. If the pinhole of a perspective camera coincides with the exterior focal point of the hyperbolic mirror, then the resulting system will have a single center of projection* COP. (b) *The geometric properties of a parabola cause all rays pointed at the focal point to be reflected parallel to the axis of the parabola. In this case, using an orthographic camera will result in a system with a single center of projection.*

Two popular approaches to wide-angle imaging are the use of rotating imaging systems and fish-eye lenses. Rotating imaging systems revolve traditional cameras about the camera pinhole and then "stitch" together the images to produce a panoramic view. Several systems have been built to this effect [6] [8]. However, for our purposes rotating systems will not help because they require a static scene, while we are interested in capturing the scene motion induced by a moving camera. On the other hand, a fish-eye camera makes use of a complex set of lenses and a very short focal length, which allows the camera to capture a hemisphere of viewing directions in a single image [9]. However, the design of a fish eye lens with a single center of projection has remained elusive. The best one can hope for is a viewpoint locus that is somewhat compact.

A third approach to wide-angle imaging is to incorporate reflecting surfaces (mirrors). These systems have been termed catadioptric systems [12]. The challenge in designing catadioptric systems is to ensure a single center of projection. Nalwa proposed aligning four planar mirrors in the shape of a pyramid [11]. The design has a single center of projection but requires the use of four cameras, one for each face of the pyramid. Yagi *et. al.* [20] and [23] used conical mirrors to produce omnidirectional images, but as shown by Nalwa [11], the design fails to have a single center of projection. Rees [17] and later Yamazawa *et. al.* [24] used a single perspective camera to image a hyperbolic mirror. When the pinhole of the cam-

era is placed at one of the foci of the hyperbola, a single center of projection is obtained (see figure 2). With careful calibration, Yamazawa *et. al.* were able to produce planar perspective images from the hyperbolic image.

In [13], Nayar and Baker investigated the class of all reflecting surfaces which have a single center of projection when imaged with a single perspective camera. This analysis led to two practical solutions, the hyperbolic and elliptic mirrors. However, by using orthographic projection, Nayar [12] showed that a parabolic mirror can also be used to achieve a single viewpoint (see figure 2). Furthermore, two parabolas can be placed back to back to capture the entire viewing sphere. Orthographic projection results in a system that requires a minimal amount of calibration, and in [16] it was shown that planar perspective images can be produced at video-rate.

Both the hyperbolic and parabolic catadioptric systems are able to capture at least a hemisphere of viewing directions about a single point of view. The single viewpoint allows us to map onto a sphere, images produced by these two systems. Likewise, knowledge of the geometry of these systems will allow us to map image velocity vectors onto a spherical representation using the Jacobian of the transformation.

3 The Motion Field on a Sphere

The motion field is the projection of the 3D velocity vector field onto a 2D surface. As pointed out by Nalwa, the motion field is an abstract geometrical concept independent of the geometry of the imaging device [10]. Therefore, we can investigate properties of the motion field using any convenient projection model. We are most familiar with the motion field equations based on the planar perspective model, because this is the most suitable model for conventional cameras. On the other hand, spherical perspective projection has the desirable property that it captures the entire motion field. This has led several researchers to investigate motion under spherical projection [4] [14] [21]. We now present the motion field equations for spherical projection and show how they can be solved using existing ego-motion algorithms.

The rigid motion of a scene point P relative to a moving camera can be described as a rotation about an axis Ω and a translation along an axis T. The instantaneous velocity of P is

$$\dot{P} = -T - \Omega \times P. \tag{1}$$

A projection model defines a function which maps scene points onto a 2D surface. In the case of spherical perspective projection the function is

$$\hat{P} = \frac{P}{\|P\|}, \tag{2}$$

where $\hat{P}$ is the projection of scene point P onto a unit sphere. The motion field equation is derived by taking the derivative of the projection function (2) with respect to time and substituting in equation (1). This leads to the following motion field equation:

$$U(\hat{P}) = \frac{1}{\|P\|}((T \cdot \hat{P})\hat{P} - T) - \Omega \times \hat{P}. \tag{3}$$

This expression describes the velocity vector $U = (\dot{x}, \dot{y}, \dot{z})^T$ at point $\hat{P} = (x, y, z)^T$ on the unit sphere as a function of rotation Ω, translation T, and scene depth $\|P\|$. The ego-motion problem is to estimate Ω and T from a set of velocity vectors U_i measured at points $\hat{P}_i$.

Three well known algorithms for estimating motion are due to Bruss and Horn [2], Zhuang *et. al.* [26], and Jepson and Heeger [5]. Although designed for a planar perspective projection, these algorithms are simple to adapt to spherical perspective projection.

Bruss and Horn: In [2], Bruss and Horn derive a depth independent constraint from the planar motion field equation and then provide a least squares solution to this constraint which is a non-linear function of T. We can remove depth from the spherical motion field equation, (3), by deriving the instantaneous epipolar constraint. Taking the cross product with $\hat{P}$ and the dot product with T, equation (3) becomes

$$T \cdot (\hat{P} \times (U + (\Omega \times \hat{P}))) = 0. \tag{4}$$

The only difference between (4) and the epipolar constraint for the planar case is that $\hat{P}$ lies on a sphere in the former and a plane in the latter. To estimate motion from a set of vectors we use the method described in [19]. A least squares estimate of Ω as a function of T is obtained and substituted back into (4), resulting in a non-linear constraint on T. An estimate of T can be found by non-linear minimization or search.

Zhuang, Huang, Ahuja, and Haralick: Zhuang *et. al.* derived a linear solution by rewriting the epipolar constraint in the form

$$a^{\mathrm{T}} h = 0,$$

where

$$a = (x^2, y^2, z^2, xy, xz, yz, y\dot{z} - z\dot{y}, z\dot{x} - x\dot{z}, x\dot{y} - y\dot{x})^{\mathrm{T}},$$

and

$$h = (l_1, l_2, l_3, 2l_4, 2l_5, 2l_6, k_1, k_2, k_3)^{\mathrm{T}}.$$

The l_i terms are functions of rotation and translation and $(k_1, k_2, k_3)^{\mathrm{T}}$ is collinear to T. Therefore the k_i provide an estimate of the direction of translation. Note that for the case of planar perspective projection $z = 1$ and $\dot{z} = 0$.

Jepson and Heeger: In [5] and [19], Jepson and Heeger describe another linear algorithm for estimating motion. They ask the question, can a linear combination of motion vectors be found which is *independent* of rotation and *orthogonal* to translation? Formally, given a set of n motion vectors at points $\hat{P}_k$, $k = 1, \ldots, n$, find coefficients c_k such that

$$(\sum_{k=1}^{n} c_k[U(\hat{P}_k) \times \hat{P}_k]) \cdot T = 0.$$

One way to find the coefficients for the spherical case is to find the null space of the following $6 \times n$ matrix,

$$\begin{bmatrix} x_1y_1 & \cdots & x_ny_n \\ x_1z_1 & \cdots & x_nz_n \\ y_1z_1 & \cdots & y_nz_n \\ x_1^2 - 1 & \cdots & x_n^2 - 1 \\ y_1^2 - 1 & \cdots & y_n^2 - 1 \\ z_1^2 - 1 & \cdots & z_n^2 - 1 \end{bmatrix}$$

The null space of this matrix is an $n-6$ dimension solution space for the coefficients. Thus, given n motion vectors, a set of $n - 6$ linear constraints on T can be found.

4 Mapping Image Motion to a Sphere

In the previous section, we demonstrated how existing methods can be used for computing ego-motion given a set of motion field vectors on a sphere. Although the image sensors discussed in section 2 are not modeled by spherical perspective projection, we can map the image motion vectors to a sphere, provided the camera system has a single center of projection. We could develop solutions to the ego-motion problem using the motion field equations for each image sensor. However, changing the representation to a sphere provides us with a general framework for developing algorithms for the entire class of wide-angle sensors, rather than tailoring each algorithm to a particular sensor.

In order to map motion in the image to motion on the sphere, we need a transformation between points in the image to points on the sphere. In addition, we need the Jacobian of this transformation in order to map the image velocities to a sphere. The transformation from image to sphere is dependent on the geometry of the imaging device. We can describe the geometry of a camera by a projection function which maps an image point to a scene ray.

Consider a coordinate system with its origin at the center of projection of the camera system. The scene rays are described in spherical coordinates (θ, ϕ) about the center of projection, where θ is the polar angle between the incoming ray and the z-axis and ϕ is the azimuth angle. The image points are described in rectangular coordinates (x, y) with the origin at the center of the image. Because the image plane is parallel to the x-y plane, ϕ in the scene will map to ϕ in the image. Therefore the relationship between ϕ and the image coordinates will be the same for all of the sensors:

$$\phi = \arctan \frac{y}{x}.$$

The projection function for the parabolic omnidirectional system is

$$\theta = 2 \arctan \frac{\sqrt{x^2 + y^2}}{h},$$

where h is the radius of the parabola in the x-y plane.

For the hyperbolic omnidirectional camera the projection function is

$$\theta = \frac{\pi}{2} - \arctan \left(\frac{(c^2 + a^2)f - 2ac\sqrt{f^2 + x^2 + y^2}}{b^2\sqrt{x^2 + y^2}} \right),$$

where a, b, and c are the parameters of the hyperbola and f is the focal length of the perspective camera.

For a fish-eye imaging system the projection function is dependent on the particular set of lenses used. The following are some of the projection functions approximated by fish-eye lenses [9]:

$$\theta = 2 \arctan \frac{\sqrt{x^2 + y^2}}{2f},$$

$$\theta = \frac{\sqrt{x^2 + y^2}}{f},$$

$$\theta = 2 \arcsin \frac{\sqrt{x^2 + y^2}}{2f}.$$

Note that fish-eye lenses do not have a single center of projection, therefore these functions are only approximations and will introduce small errors dependent on how compact the center of projection is for the fish-eye lenses used in practice. This underscores the importance of designing wide-angle imaging systems with a single center of projection.

For spherical perspective projection the scene ray (θ, ϕ) will map to the point (θ, ϕ) on the sphere, therefore the projection functions provide a mapping from

image space to spherical projection. To map image velocity vectors to a sphere it will *not* suffice to project the endpoints of the vector using the projection functions. To properly transform velocity vectors we must use the Jacobian of the transformation. The Jacobian relates partial derivatives in one coordinate system to those in another. Each sensor has its own Jacobian which can be derived by taking the derivative of the projection function with respect to image coordinates. We do this only for the parabolic sensor but the method is the same for all of the projection functions.

$$J = \begin{bmatrix} \frac{\partial\theta}{\partial x} & \frac{\partial\theta}{\partial y} \\ \frac{\partial\phi}{\partial x} & \frac{\partial\phi}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{2hx}{\sqrt{y^2+x^2}(h^2+y^2+x^2)} & \frac{2hy}{\sqrt{y^2+x^2}(h^2+y^2+x^2)} \\ -\frac{y}{y^2+x^2} & \frac{x}{y^2+x^2} \end{bmatrix}$$

The velocity vector in image space is the rate of change of image coordinates with respect to time,

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix}$$

Transforming the image velocity vector by the Jacobian produces,

$$\begin{bmatrix} \frac{d\theta}{dt} \\ \frac{d\phi}{dt} \end{bmatrix} = J \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix},$$

which is a measure of angular velocity. Equation (3), the motion vector for a point on the sphere, is the velocity vector tangent to the motion on the sphere. Therefore, we need one more transformation S,

$$S = \begin{bmatrix} \frac{\partial x}{\partial\theta} & \frac{\partial x}{\partial\phi} \\ \frac{\partial y}{\partial\theta} & \frac{\partial y}{\partial\phi} \\ \frac{\partial z}{\partial\theta} & \frac{\partial z}{\partial\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\phi & -\sin\theta\sin\phi \\ \cos\theta\sin\phi & \sin\theta\cos\phi \\ -\sin\theta & 0 \end{bmatrix},$$

which takes us from angular to rectangular velocities and is independent of the sensor used. Now, velocity U on a sphere can be found by the following transformation of image velocity:

$$U = SJ \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix}.$$

5 Experiments

We use the following method for recovering egomotion from a series of images taken by the parabolic omnidirectional sensor described in [12]. First a set of images is obtained while the camera is moving. From the image set, image velocity (optic flow) is computed.

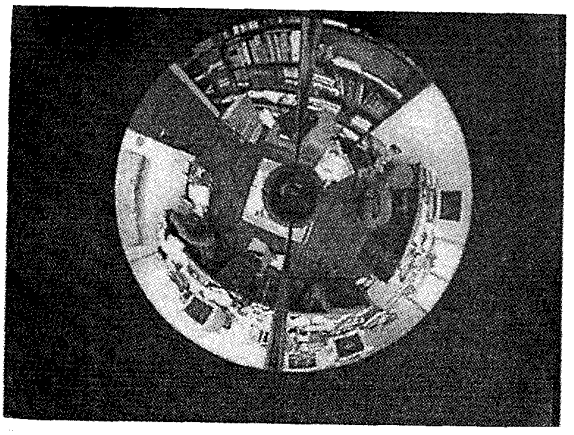

Figure 3: *An image taken by the parabolic omnidirectional camera. This is the scene where the motion experiments were conducted.*

Then, the velocity vectors are mapped to a sphere using the Jacobians described in the preceding section. Once the velocity vectors are projected onto a sphere, ego-motion is estimated using the three algorithms described in section 3.

To compute optic flow in the omnidirectional images we use the gradient based method of Lucas and Kanade [7] with the extensions of Simoncelli *et. al.* [18]. The code was obtained from the Barron *et. al.* optic flow survey [1]. Care must be taken when selecting an optic flow algorithm for computing image velocity in an omnidirectional image. Omnidirectional cameras introduce non-affine distortions into the image, a consequence of projecting a hemisphere of viewing directions onto a planar image. Therefore, correlation based algorithms that do not account for the distortions are likely to fail.

When performing motion experiments accurate ground truth is difficult to obtain. However, a catadioptric camera has an exterior center of projection, which allows the camera to be manually centered over a rotation stage. To obtain a measure of error for the estimated direction of translation, we use the following method. In each experiment, the camera undergoes a motion (translation and/or rotation)and a sequence of images is taken. From this sequence the direction of translation, T_1, is estimated. Then the camera is rotated α° and the same motion is applied and the direction of translation, T_2 is estimated. The error is measured using the following formula:

$$\text{error} = |\alpha^\circ - \arccos(T_1 \cdot T_2)|. \quad (5)$$

In the first experiment the camera undergoes a pure

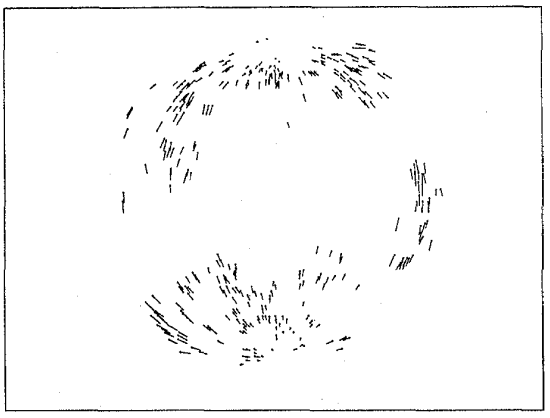

Figure 4: *Omnidirectional optic flow (mag. $\times 20$) for translational motion. Because the camera is translating parallel to the parabolic mirror, both the focus of expansion and contraction exist in the image.*

translation of 0.5 cm. per frame. Then the camera is rotated $\alpha = 10°$ and the same translational motion is applied. Ideally, the distance between the two computed translational motions is 10°. The results of this experiment are shown in figure 5.

The optic flow algorithm only computes optic flow vectors, where the confidence measure is above the threshold, τ. As τ is increased the number of vectors decreases. Results are shown for several different τ values.

τ	3.0	3.5	4.0	4.5
Bruss-Horn	1°	1°	1°	1°
Heeger-Jepson	3.14°	2.12°	0.9°	1.71°
Zhuang	3.14°	2.12°	0.9°	1.71°

Figure 5: *The error in estimating the direction of translation for purely translational motion. τ is a threshold used by the optic flow algorithm [1]. The fluctuation of the error as τ changes, provides a measure of stability.*

In the second experiment the camera undergoes a translation of 0.5 cm. per frame and a rotation of 0.5° per frame. Again the camera is rotated $\alpha = 10°$ and the same translation and rotation is applied. The results are shown in figure 7. To solve the non-linear minimization problem (Bruss-Horn) a coarse to fine search is used. The final search is performed at 1° increments. Therefore, 1° provides a bound on the accuracy of the non-linear algorithm. The data reflects the fact that the non-linear algorithm is both more accurate and more stable than the linear algorithms.

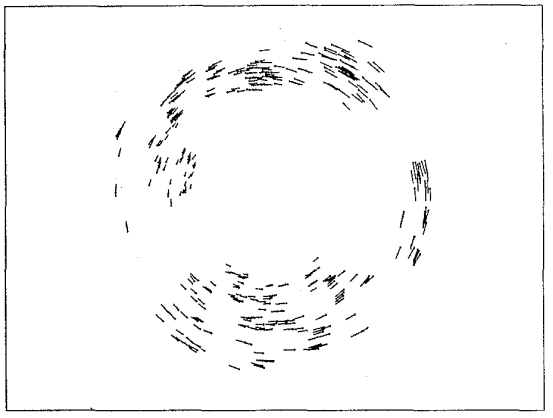

Figure 6: *Omnidirectional optic flow (mag. $\times 10$) for general motion (translation and rotation). Note that the rotational motion has obscurred the direction of translation.*

τ	3.0	3.5	4.0	4.5
Bruss-Horn	1°	2°	1°	2°
Heeger-Jepson	1.34°	4.46°	8.61°	7.16°
Zhuang	1.34°	4.45°	8.61°	7.16°

Figure 7: *The error in estimating the direction of translation for general motion (rotation and translation).*

6 Conclusion

Recent research in wide-angle imaging has produced camera systems with fields of view greater than a hemisphere. In addition, research in motion estimation has provided a number of algorithms to compute ego-motion. Although designed for planar perspective cameras, these algorithms can be adapted to omnidirectional cameras by mapping the optic flow field to a sphere, via the use of an appropriate Jacobian. We have shown how this mapping can be performed for a variety of wide-angle cameras.

References

[1] J. Barron, D. Fleet, S. Beauchemin and T. Burkitt. "Performance of optic flow techniques," IJCV, vol. 5, pp. 77-104.

[2] A. Bruss and B.K.P. Horn. "Passive Navigation," Computer Vision, Graphics, and Image Processing, vol. 21, pp. 3-20, 1983

[3] D. Coombs, M. Herman, T. Hong and M. Nashman, "Real-time obstacle avoidance using central flow divergence and peripheral flow," ICCV '95, pp. 276-283.

[4] C. Fermuller and Y. Aloimonos. "Direct Perception of Three-Dimensional Motion from Pattern of Visual Motion," Science, vol. 270, pp. 1973-1976, 1995.

[5] A.D. Jepson and D.J. Heeger. "A fast subspace algorithm for recovering rigid motion," Proc. of IEEE Workshop on Visual Motion, pp. 124-131, 1991.

[6] A. Krishnan and N. Ahuja. "Panoramic Image Acquisition," CVPR '96, pp. 379-384, June 1996.

[7] B. Lucas and T. Kanade. "An iterative image restoration technique with an application to stereo vision," DARPA IU Workshop '81, pp. 121-130.

[8] L. McMillan and G. Bishop. "Plenoptic Modeling: An Image-Based Rendering System," Computer Graphics: Proc. of SIGGRAPH '95, pp. 39-46, August 1995.

[9] K. Miyamoto. "Fish Eye Lens," Journal of Optical Society of America," 54(8):1060-1061, August 1994.

[10] V. Nalwa. A Guided Tour of Computer Vision, Addison-Wesley, Reading, Massachusetts, 1993.

[11] V. Nalwa. "A True Omnidirectional Viewer," Tech. report, Bell Laboratories, Holmdel, NJ 07733, U.S.A., Feb. 1996.

[12] S. K. Nayar. "Catadioptric Omnidirectional Camera," CVPR '97, pp.482-488.

[13] S. K. Nayar and S. Baker. "Catadioptric image formation," DARPA IU Workshop '97, New Orleans, May 1997.

[14] R. C. Nelson and J. Aloimonos. "Finding Motion Parameters from Spherical Motion Fields (Or the Advantages of Having Eyes in the Back of Your Head)," Biological Cybernetics, vol. 58, pp. 261-273, 1988.

[15] S. Oh and E. Hall. "Guidance of a Mobile Robot using an Omnidirectional Vision Navigation System," Proc. of the Society of Photo-Optical Instrumentation Engineers, 852:288-300, November 1987.

[16] V. Peri and S. K. Nayar. "Omnidirectional Video Processing," Proc. of IROS U.S.-Japan Graduate Student Forum on Robotics, November 1996.

[17] D. W. Rees. "Panoramic Television Viewing System," U.S. Patent No. 3,505,465, 1970.

[18] E. Simoncelli, E. Adelson, and D. Heeger. "Probability distributions of optical flow," CVPR '91, pp. 310-315.

[19] T. Tian, C. Tomasi, and D. Heeger. "Comparison of Approaches to Egomotion Computation," CVPR '96.

[20] Y. Yagi and S. Kawato. "Panoramic Scene Analysis with Conic Projection," Proc. of Int'l Conference on Robots and Systems (IROS), 1990.

[21] B. Yen and T. S. Huang. "Determining 3-D Motion and Structure of a Rigid Body Using the Spherical Projection," CVIP, vol. 21, pp. 21-32, 1983.

[22] Y. Yagi, W. Nishii, K. Yamazawa and M. Yachida. "Rolling Motion Estimation for Mobile Robot by Using Omnidirectional Image Sensor HyperOmniVision," Proc. of Int'l Conference on Pattern Recognition '96, pp. 946-950, 1996.

[23] Y. Yagi, S. Kawato and Y. Tsuji. "Collision Avoidance using Omnidirectional Image Sensor (COPIS). Proc. of Int'l Conf. on Robotics and Automation, May 1991.

[24] K. Yamazawa, Y. Yagi and M. Yachida. "Omnidirectional Imaging with Hyperboloidal Projection," Proc. of the Int'l Conf. on Robots and Systems (IROS), 1993.

[25] K. Yamazawa, Y. Yagi and M. Yachida. "Obstacle Avoidance with Omnidirectional Image Sensor HyperOmni Vision," Proc. of the IEEE Int'l Conf. on Robotics and Automation, pp. 1062-1067, May 1995.

[26] X. Zhuang, T.S. Huang, N. Ahuja and R.M. Haralick. "A Simplified Linear Optic Flow- Motion Algorithm," Computer Vision, Graphics, and Image Processing, vol. 42, pp 334-344, 1988.