Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

A Human Eye Like Perspective for Remote Vision

Curtis M. Humphrey, Stephen R. Motter, and Julie A. Adams
Depart. of Electrical Engineering and Computer Science
Vanderbilt University
Nashville, TN, USA
[curtis.m.humphrey, stephen.r.motter, julie.a.adams]@vanderbilt.edu

Mark Gonyea
Math and Computer Science
Vanderbilt RET and Smyrna High School
Smyrna, TN, USA
gonyeam@rcs.k12.tn.us

***Abstract*— Robots in remote environments (e.g., emergency response) have many potential benefits and affordances, with imagery (or video) being a major, if not primary, affordance. However, remote imagery is usually affected by the keyhole effect, or viewing the world through a "soda straw." This work focuses on reducing the keyhole effect by improving the viewing angle of the imagery using a novel method that produces results more akin to that provided by the human vision system. The method and early results are subsequently presented for this human eye like perspective for remote vision.**

***Keywords*—keyhole effect, remote vision, human eye like perspective, teleoperation**

I. INTRODUCTION

Robots in remote environments (e.g., emergency response) have many potential benefits [1] and affordances, with imagery (or video) being a major, if not primary, affordance. Imagery provides two primary benefits: scene awareness/observation and navigation. Scene awareness and observation allows human responders the ability to understand and create mental models of the remote environment that will assist them in a wide range of tasks [2, 3]. The second benefit, navigation, is important even in situations where the robots are designed to autonomously navigate. There are many navigational challenges present in emergency responses [4] that can cause the autonomous navigation to fail, thereby requiring a remote operator to teleoperate the robot (i.e., use imagery for direct navigation). Success in teleoperation is often dependent on the operator's situational awareness (i.e. understanding of the robot's immediate and large-scale environment) [5].

These two benefits, scene awareness/observation and navigation, are best performed, however, with *different* viewpoints. Wickens and Prevett [6] have shown that ego- and near exo-referenced viewpoints are better for navigation and far exo- and world-referenced viewpoints are better for understanding the situation or scene awareness/observation. Others have shown that the near exo-reference viewpoint is better than ego-reference for teleoperation of robots [7].

Even with the near exo-reference viewpoint, there remain issues relating to the keyhole effect. The keyhole effect occurs when one views the world through a narrow field of view, such as viewing the world through a "soda straw" [8]. The main cause of the keyhole effect is that the natural dynamic relationship between the human perceptual system and the scene are decoupled [9]. Alternatively, the remote camera on the robot does not provide all the affordances of a real human eye. The three main affordances that are usually not provided are viewing angle [10], saccades or rapid eye motion [11, 12], and stereoscopic vision [10]. The stereoscopic vision affordance is not as important for understanding the scene, as the human vision system only accommodates (i.e., adapts) within the first twenty feet [12] and after that the error between perceived distance and physical distance increases greatly [13]. The other two affordances are important for scene understanding.

A. Viewing Angle

The human eye has a 200-degree viewing angle [11]; however, the detail, or visual resolution, across this arc is not constant. The eye has two viewing elements: rods, for light and dark vision, and cones, for color vision [12]. The distribution of these elements varies with angular distance from the center of the eye so that the center of the eye is responsible for details (e.g. reading text) and the periphery is responsible for context (see Fig. 1).

There have been several approaches to break or reduce the keyhole effect by increasing the viewing angle. The solutions generally follow one of four approaches. One approach has been to change the viewpoint reference to near exo-referenced; thereby effectively increasing the viewing angle [7]. Another approach has been to use multiple cameras to create a perspective folding view (i.e, one camera viewing the center and four cameras viewing the edges in the shape of a plus "+") [8]. A third approach has been to use a fish-eye lens [14], while the fourth approach employs an omnidirectional lens [15]. Although these approaches have increased the viewing angle and have been found to be useful in certain contexts, none of these approaches provides the same perspective affordances as the human eye. The near exo-referenced view and the perspective folding view provide the same image detail across the entire arc, which, if increased to 200-degrees may result in issues such as motion sickness [10]. The fish-eye view and the omnidirectional view provide varying degrees of image detail across the arc, but in a distorted space that is unlike that

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

provided by the human eye. The human eye center vision is relatively undistorted, it is the periphery vision that is distorted and provides less detail [12] (Fig. 2). Both fish-eye and omnidirectional views do not provide this undistorted center vision. A fish-eye view may appear to provide an undistorted center area; however, it is still radial distorted (i.e. a horizontal line above the center point will appear as a curve line in the center area and not a horizontal line).

Figure 1: How approximately the human eye sees a scene [12].

The concept of a human eye-like perspective builds on the perspective folding view concept [8]. The perspective folding view increases the viewing angle; however, it achieves this increase by using five cameras. A limitation is that the five cameras may require more bandwidth than is possible to provide in remote environments [7]. Furthermore, the resulting image provides the same level of detail across the entire arc, which is fundamentally different from the way that the human eye samples the world. This paper focuses on increasing the viewing angle in a manner that more closely approximates the perspective provided by the human eye.

This paper focuses on addressing the viewing angle affordance by proposing a method for presenting a human eye like perspective. The following sections present the details of this approach, provide resulting images, and discuss findings and future work.

II. METHOD

The human eye like perspective method combines two images viewing the same point in space from the same, or approximately the same, point in space using two different focal lengths into one coherent image. Although this method currently uses two cameras or images, the resulting image can also be achieved using a single, albeit complex, lens on one camera. The two views (wide angle and telephoto angle) are merged together using a transformation function that results in a final image with a relatively undistorted focus or detailed center and a surrounding context or peripheral area, thereby simulating the perspective provided by the human eye. This paper explores three different transform functions with different scale values.

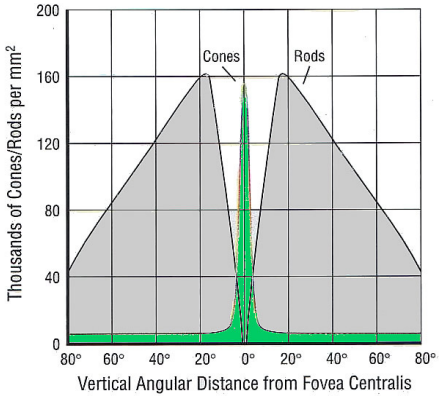

Figure 2: Cones and rod vertical density graph depicting the concentration of viewing cells by angular distance in the human eye [12].

The image with the smaller focal length (i.e., greater field of view) is henceforth called the *context image* (cI), as this image provides the periphery or context region in the final image. The image with the greater focal length (i.e., smaller field of view) is henceforth called the *detail image* (dI), as this image provides the center or detailed region of the final image.

The algorithm is composed of three steps: align, transpose, and merge. The align step transforms the context and detail images into a common coordinate system, that is, a pixel representing the same physical real-world location has the same χ and y coordinates in both images. The transpose step maps the pixels from the context and detail images into a pre-final image matrix. The third step, merge, transforms the pre-final image matrix into the final image by resolving two issues in the pre-final image: locations with more than one pixel and locations with no pixels.

The resulting images, presented in Section III, are 800 by 600 pixel images constructed from two 320 by 240 pixel images. Each two image set, detail and context, were taken by the same digital camera from the same location on a tripod using two different focal lengths, 6mm and 17mm.

A. Align Step

The align step requires two functions. The first function transforms the coordinates of the context image such that each unit represents the same physical space as one unit in the detail coordinate system. For example, if the context image represents twice the physical width as the detail image, the context image will have its coordinates doubled (i.e., if x was 2 it is now 4). The second function shifts the coordinates of the detail image such that the physical space represented by the first pixel in the detail image has the same x and y location as the same physical space in the context image after it has been transformed. For example, if both images are focused on the same location in space, the detail image will have its coordinates shifted so that its center has the same coordinates as the recently transformed context image center coordinates.

B. Transpose Step

The purpose of the transpose step is to compress the periphery of the aligned images while minimizing the distortions in the center or detail section of the final image. This research explored three transpose functions: linear, sinusoidal, and Gaussian. All three transpose functions followed the same form; that is, each mapped an aligned coordinate (e.g. x) into a final coordinate given a scaling factor (S_{factor}). The scaling factors were chosen to exemplify the variation range possible within each transform type. Each pixel

from both the context and detail images was copied to the pre-final image matrix based on the mapping of their aligned x and y coordinates into the final coordinates. Throughout this section, the equations are depicted in terms of the x-axis; however, the equations are also used on the y-axis by substituting y for x and height for width.

1) Common Elements of Linear and Sinusoidal

The linear and sinusoidal transpose functions compute the new coordinate locations using the same basic procedure. Both divide the axis into three parts based on the focal length ratio (f_{ratio}) between the context image and the detail image (1).

$$f_{ratio} = \frac{focal\_length_{dI}}{focal\_length_{cI}} \quad (1)$$

The three parts are defined by the two points, $part_1$ and $part_2$, as defined in (2) and (3). The x_{offset} term is used to adjust the position of the detail image relative to the center of the context image for cases where the two images are not taken from the same location (i.e., not co-located).

$$part_1 = \frac{width_{cI} * f_{ratio} - width_{dI}}{2} + x_{offset} \quad (2)$$

$$part_2 = part_1 + width_{dI} \quad (3)$$

Both the linear and the sinusoidal transpose functions use a compression ratio (C_{ratio}) for transforming the aligned images coordinates into the final image coordinates (4).

$$C_{ratio} = \frac{width_{final\_image} - width_{dI}}{width_{cI} * f_{ratio} - width_{dI}} \quad (4)$$

2) Linear Transpose Function

The linear transpose function computes the final image coordinate based on the aligned image coordinate and a scaling factor (S_{factor}), which can range from 1.0 to 1.5. The linear transpose function is defined in (5). When the scaling factors are 1.0 and 1.5 the align coordinates map to the final coordinates as depicted in Fig. 3.

3) Sinusoidal Transpose Function

The sinusoidal transpose function computes the final image coordinate based on the aligned image coordinate and a scaling factor (S_{factor}), which can range from 1.0 to 1.5. This function uses a corrected arc sine function (CAS) that returns a continuous value from zero to one based on the value of x and the $width$ of the current part (8).

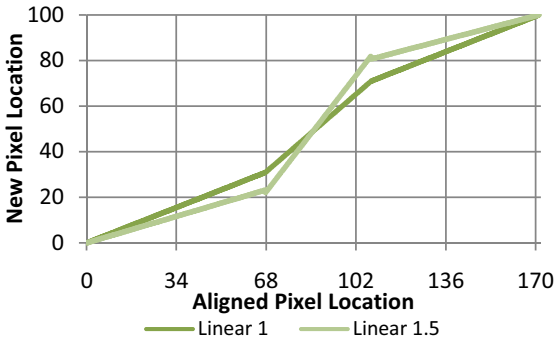

Figure 3: New Pixel Location (y-axis) vs. Aligned Pixel Location (x-axis) for the linear transpose function.

$$CAS(x, width) = sin^{-1}\left(2 * \left(\frac{x}{width}\right) - 1\right) * \frac{1}{\pi} + \frac{1}{2} \quad (8)$$

The sinusoidal transpose function is defined in (6). When the scaling factors are 1.2 and 1.42 the align coordinates map to the final coordinates as depicted in Fig. 4.

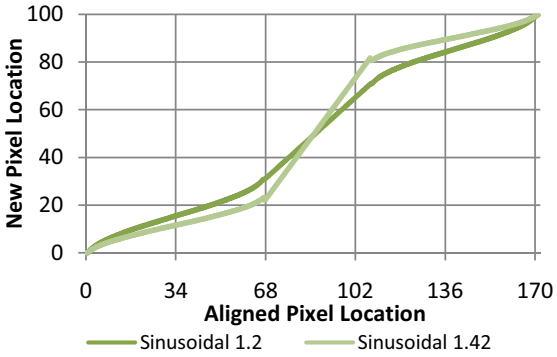

Figure 4: New Pixel Location (y-axis) vs. Aligned Pixel Location (x-axis) for the sinusoidal transpose functions.

$$\text{Linear}(x, S_{factor}) = \begin{cases} x * \frac{C_{ratio}}{2 - 1/S_{factor}}, & x < part_1 \\ x * S_{factor} + \frac{width_{cI} * f_{ratio}}{2}(1 - S_{factor}) - part_1(1 + C_{ratio}), & part_1 \le x \le part_2 \\ (x - part_2 + part_1) * \frac{C_{ratio}}{S_{factor}} + width_{dI} * S_{factor}, & x > part_2 \end{cases} \quad (5)$$

$$\text{Sinusoidal}(x, S_{factor}) = \begin{cases} ASC(x, part_1) * part_1 * \frac{C_{ratio}}{2 - 1/S_{factor}}, & x < part_1 \\ x * S_{factor} + \frac{width_{cI} * f_{ratio}}{2}(1 - S_{factor}) - part_1(1 + C_{ratio}), & part_1 \le x \le part_2 \\ (CAS(x - part_2, part_1) + 1) * part_1 * \frac{C_{ratio}}{S_{factor}} + width_{dI} * S_{factor}, & x > part_2 \end{cases} \quad (6)$$

$$Gaussian(x, S_{factor}) = \frac{Location(x, S_{factor}) - Location(1, S_{factor})}{Location(width_{cI} * f_{ratio}, S_{factor}) - Location(1, S_{factor})} * width_{final\_image} \quad (7)$$

4) Gaussian Transpose Function

The Gaussian transpose function computes the final image coordinate based on the aligned image coordinate and a scaling factor (S_{factor}), which has a useful range from 1 to 12. This function uses a location function (9), which is based on an approximation of the cumulative distribution function (CDF) as defined by the Abromowitz and Stegun algorithm 26.2.17 [16].

$$Location(x, \mathrm{s}) = CDF\left(\left(\frac{2*x}{width_{cI}*f_{ratio}-1} - 1\right) * \mathrm{s}\right) \quad (9)$$

The Gaussian transpose function is defined in (7).When the scaling factors are 1.667, 2.5, and 3.124 the align coordinates map to the final coordinates as depicted in Fig. 5.

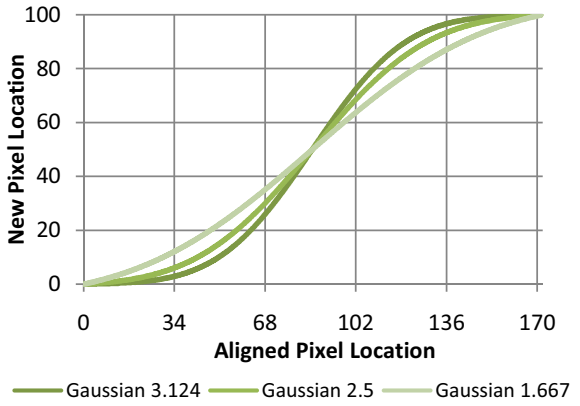

Figure 5: New Pixel Location (y-axis) vs. Aligned Pixel Location (x-axis) for the Gaussian transpose functions.

C. Merge Step

The purpose of the merge step is to transform the pre-final image matrix into the final image. The transformation involves two steps: averaging pixel locations that have more than one pixel, and filling in pixel locations that have no pixels. Fig. 6 depicts how each transform function maps more than one pixel to certain final image coordinate locations. Notice that in Fig. 6 the three transpose functions are similar to Fig. 1 in that the center region (i.e., from locations 35 to 65) has the most details (i.e., lowest number of combined pixels) and the periphery (i.e., from locations 0 to 35 and 65 to 100) have fewer details (i.e., higher number of combined pixels). Thus, all three methods provide a different approximation of the human eye perspective. The Gaussian transpose function is, however, the closest match to the human eye detail distribution (Fig. 1). A straightforward approach was taken that averaged all pixels, if there were more than one, to produce the final image pixel. For final image coordinate locations that did not have at least one pixel, the final image pixel was computed by averaging the nearest neighbors with pixel values.

III. Results

The three transpose functions, linear, sinusoidal, and Gaussian, were tested on many image sets with many scaling factor (S_{factor}) values. The resulting image sets were shown to six people and a consensus was formed as to which transpose

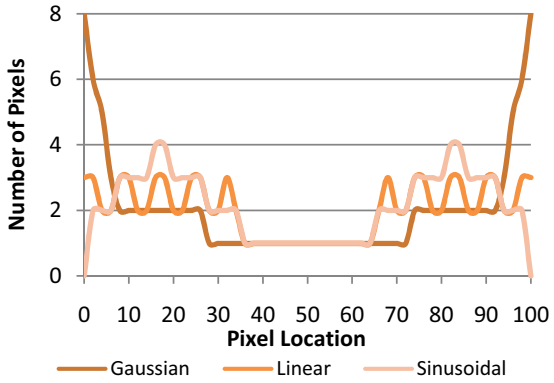

Figure 6: Number of pixels mapped to each final location that then have to be combined when the context, detail, and final image are all 100x100 pixels for linear (1.5), sinusoidal (1.42), and Gaussian (2.5) transpose functions.

functions and scale values provided the best details to context ratio with the most understandable distortion in the periphery section. The two combinations determined to have the most interesting potential were the sinusoidal with a 1.42 scaling factor and the Gaussian with a 2.5 scaling factor. The two image sets are provided to illustrate how the different transpose functions combine the detail and context images into the final image.

A. Image Set A

The first image set, A, depicts a possible situation where a ground robot is exploring an indoor room and is inspecting a collection of wires (Fig. 7). This set illustrates a situation where the images are employed to provide details on a focus area (i.e., wires) as well as context in the periphery the final image. The linear transpose function result is provided to depicted the combined image without any distortion (i.e., S_{factor} = 1.0) (Fig. 8). When comparing the linear (Fig. 8), sinusoidal (Fig. 9), and Gaussian (Fig. 10) resulting images, both the sinusoidal and Gaussian functions provide a larger context area that contains more detail than that provided by the linear function. This larger, minimally distorted context area, as compared to the periphery area, is the primary difference between the human eye like perspective and other methods. The primary difference between the sinusoidal and Gaussian final images is the type of distortion in the periphery area. When comparing the pipes at the bottom of the images in Figs. 9 and 10, the Gaussian distorts the periphery in a manner that the floor is hardly visible (i.e., represented by only a few pixels) whereas the sinusoidal distorts the periphery so that the floor is more visible. The tradeoff is that the sinusoidal results in more distortion of the space midway between the context and the edge of the picture; however, in this image set (i.e., A) that distortion is not as obvious.

B. Image Set B

The second image set, B, depicts a view that may appear from an aerial robot as it navigates through a corridor while following a person on the ground (e.g., a first responder) (Fig. 11). This image set shows how the image can provide minimally distorted details of the person while still providing the context of the robot's location relative to the sidewalls. The linear transpose function result is provided to depicted the

Figure 7: The context (left) and detail (right) images used in set A.

Figure 8: Set A Linear transposed with S_{factor} = 1.0 (i.e., no distortion).

Figure 9: Set A Sinusoidal transposed with S_{factor} = 1.42.

Figure 10: Set A Gaussian transposed with S_{factor} = 2.5.

Figure 11: The context (left) and detail (right) images used in set B.

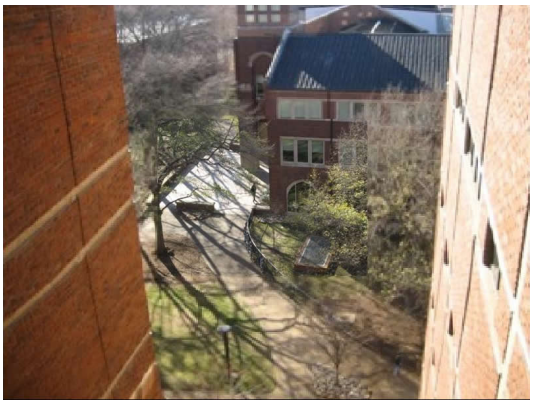

Figure 12: Set B Linear transposed with S_{factor} = 1.0 (i.e., no distortion).

Figure 13: Set B Sinusoidal transposed with S_{factor} = 1.42.

Figure 14: Set B Gaussian transposed with S_{factor} = 2.5

combined image without any distortion (i.e., S_{factor} = 1.0) (Fig. 12). When comparing the linear (Fig. 12), sinusoidal (Fig. 13), and Gaussian (Fig. 14) resulting images, both the sinusoidal and Gaussian functions provide an image with a larger context area resulting in the person appearing larger, especially with the Gaussian. The primary difference between the sinusoidal and Gaussian final images is the type of distortion in the periphery area, as was the case for set A. When comparing the building lines on the left side of the images in Figs. 13 and 14 it becomes noticeable that the sinusoidal keeps the lines straighter for a longer section than the Gaussian. Although the lines are straighter in the sinusoidal they are not at the same angle as the lines in the linear image (Fig. 12) because the sinusoidal is still distorting the space to provide more area for the context.

IV. CONCLUSION

This initial work was a success, as the resulting images (Figs. 9, 10, 13, 14) do provide a view of the situation that is different from other solutions to increase the view angle and the results do not appear to be as cognitively confusing. Upon reviewing the resulting images from the transpose functions (i.e., linear, sinusoidal, and Gaussian) over a wide range of image sets with different scaling factors, two combinations have the most interesting potential: the sinusoidal with a 1.42 scaling factor and the Gaussian with a 2.5 scaling factor. The preferred transpose function, when reviewing the results from image sets A and B, was the sinusoidal with a 1.42 scaling factor as it presents the best tradeoff between providing a clear detail area and a minimally distorted, but still understandable periphery or context area (Figs. 9 and 13). Future work will focus on different focal lengths of the context and detail images as well as performance through a variety of tasks using video, rather than still images. Future work will also include human subject image quality evaluations of the resulting images.

ACKNOWLEDGMENT

This work was supported by the NSF Grants IIS-0519421, EEC-0742871, and EEC-0338092. The authors thank Dr. Stacy Klein for organizing the Vanderbilt Research Experience for Teachers (RET) program.

REFERENCES

[1] C. M. Humphrey and J. A. Adams, "Robotic Tasks for CBRNE Incident Response," *Advanced Robotics*, in press. .

[2] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. M. Humphrey, "Supporting wilderness search and rescue using a camera-equipped mini UAV: Research Articles," *Journal of Field Robotics*, vol. 25, 2008, pp. 89-110.

[3] J. A. Adams, C. M. Humphrey, M. A. Goodrich, J. L. Cooper, B. S. Morse, C. Engh, and N. Rasmussen, "Cognitive Task Analysis for Developing UAV Wilderness Search Support," *Journal of Cognitive Engineering and Decision Making*, vol. 3, 2009, pp. 1-26.

[4] J. Casper and R. R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 33, 2003, pp. 367-385.

[5] J. Scholtz, J. Young, J. Drury, and H. A. Yanco, "Evaluation of human-robot interaction awareness in search and rescue," *Proceedings of IEEE International Conference on Robotics and Automation*, 2004, pp. 2327-2332.

[6] C. D. Wickens and T. T. Prevett, "Exploring the dimensions of egocentricity in aircraft navigation displays," *Journal of Experimental Psychology: Applied*, vol. 1, 1995, pp. 110-135.

[7] B. Keyes, R. Casey, H. A. Yanco, B. A. Maxwell, and Y. Georgiev, "Camera Placement and Multi-Camera Fusion for Remote Robot Operation," *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, Gaithersburg, MD: National Institute of Standards and Technology (NIST), 2006.

[8] M. Voshell, D. D. Woods, and F. Phillips, "Overcoming the Keyhole in Human-Robot Coordination: Simulation and Evaluation," *Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 49, 2005, pp. 442-446.

[9] J. S. Tittle, A. Roesler, and D. D. Woods, "The Remote Perception Problem," *Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 46, 2002, pp. 260-264.

[10] J. Chen, E. Haas, and M. Barnes, "Human Performance Issues and User Interface Design for Teleoperated Robots," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, 2007, pp. 1231-1245.

[11] "Human eye - Wikipedia, the free encyclopedia," Mar. 2009.

[12] T. R. Quackenbush, *Relearning to See: Improve Your Eyesight -- Naturally!*, Berkeley, CA, USA: North Atlantic Books, 2000.

[13] H. S. Smallman and M. St. John, "Naive Realism: Misplaced Faith in Realistic Displays," *Ergonomics in Design*, vol. 13, Summer. 2005, pp. 6-13.

[14] S. Shah and J. Aggarwal, "Mobile robot navigation and scene modeling using stereo fish-eye lens system," *Machine Vision and Applications*, vol. 10, Dec. 1997, pp. 159-173.

[15] K. Yamazawa, Y. Yagi, and M. Yachida, "Omnidirectional imaging with hyperboloidal projection," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1993, pp. 1029-1034.

[16] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*, New York, NY: Dover Publications, 1964.