Proceedings of the 2004 IEEE
International Conference on Networking, Sensing & Control
Taipei, Taiwan, March 21-23, 2004

Construction of Perspective And Panoramic Images from Omni-images Taken from Hypercatadioptric Cameras for Visual Surveillance

Sheng-Wen Jeng
Department of Computer and Information Science
National Chiao Tung University
1001 Ta Hsueh Road, Hsinchu, Taiwan 300,
Republic of China
sunny@.itri.org.tw

Wen-Hsiang Tsai
Department of Computer and Information Science
National Chiao Tung University
1001 Ta Hsueh Road, Hsinchu, Taiwan 300,
Republic of China
whtsai@cis.nctu.edu.tw

Abstract - *A novel method for construction of perspective and panoramic images from omni-images taken from hypercatadioptric cameras for visual surveillance is proposed. The constructing work includes two major steps: unwarping of the omni-image into a rectangularly tessellated perspective or panoramic image, followed by interpolation of the coordinates of the image pixels which are unfilled in the first step. For the former step, techniques for combining the uses of a back-projection method and some geometric relations within the camera system to yield a forward-projection effect to accomplish the unwarping task are employed. And for the latter step, a novel scheme, called 8-directional-regions interpolation, to interpolate the unfilled pixels' coordinates in the image yielded in the first step to complete the image construction work is proposed. Good experimental results are also included to show the feasibility of the proposed method.*

Keywords: omni-image, hypercatadioptric camera, non-single-viewpoint design, perspective image, panoramic image, 8-directional-regions interpolation.

1 Introduction

For a visual surveillance system, expanding the field of view (FOV) of the system to enhance the visual coverage and to reduce the dead corners of the system monitoring scope is one of the most important issues in visual surveillance studies [1, 2]. In recent years, an *omnidirectional camera*, called simply *omni-camera* in the sequel, was used frequently for the purpose of visual surveillance, aiming at expanding the FOV of the camera to cover the entire monitoring scope. An image taken by an omni-camera is called an omnidirectional image, or simply an *omni-image*. Such a kind of image is usually highly distorted and is not suitable for human observation. Figure 1 shows an omni-image taken by an omni-camera used in this study. Usually, part of the omni-image is reconstructed into a rectangularly tessellated *perspective image* for the purpose of comfortable observation. Figure 2 shows an example of a constructed perspective image of Figure 1.

Limited by the nature of the perspective image, the FOV of an image of this kind is usually confined to a narrow range. When considering the need of comfortable observations and the elimination of the dead corners of the system monitoring scope, the panoramic image, whose corresponding FOV is 360 degrees, is frequently adopted for use in a visual surveillance system. Figure 9(c) shows an example of a constructed panoramic image.

An omni-camera is a system resulting from an integration of a CCD sensor chip, a convex reflection mirror, and a projection lens. In this study, a combination of a hyperbolic mirror and a perspective lens is used to construct a camera system, called a *hypercatadioptric camera*. Usually, a hypercatadioptric camera is designed to fit the *single-view-point (SVP) constraint* [3-5] for ease to carry out the unwarping work. But it is possible that an SVP-designed camera was not constructed precisely enough so that the camera becomes a non-SVP-constrained one. On the other hand, for the purpose to expand the FOV or to increase the image resolution, sometimes it might be desired as well to design a non-SVP-constrained hypercatadioptric camera. Foreither case, the image unwarping process is very complicated. In our previous work [6], we have developed a set of general analytic equations for modeling the back-projection of a point in the omni-image into a point in the real world for a non-SVP-designed hypercatadioptric camera. To construct a perspective view image like Figure 2 or a panoramic image like Figure 9(c), two major steps are proposed in this study, namely, unwarping of the omni-image into a rectangularly tessellated perspective or panoramic image, either called an *unwarped image* in the sequel, followed

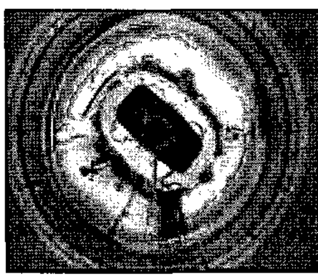
Figure 1: An omni-image taken by an omni-camera used in this study.


Figure 2: A reconstructed perspective image of Figure 1.

by interpolation of the coordinates of the image pixels which are *unfilled* in the unwarped image. An example of an unwarped image with unfilled pixels is shown in Fig. 12(a), which, being irregularly distributed, deteriorates the image quality. For the first step, a scheme has been designed by Jeng and Tsai [6] using the previously mentioned analytic back-projection equations to get a forward-projection effect. And to deal with the problem of *interpolation at unfilled pixels*, which are different from that of common image transformation operations like image scaling, rotation, etc., a so-called *8-directional-regions interpolation* method is proposed in this study to accomplish the image construction work.

The remainder of this paper is organized as follows. In Section 2, we describe the proposed process for constructing perspective or panoramic images from omni-images taken from a non-SVP designed hypercatadioptric camera. In Section 3, we describe briefly an example of visual surveillance systems, using a non-SVP designed hypercatadioptric camera and a personal computer. In Section 4, we show some experimental results, and make some conclusions finally in Section 5.

2 Image Construction

For an SVP-designed hypercatadioptric camera, unwarping a perspective image from an omni-image is a process of forward-projection from a point $X_p(x_p, y_p, z_p)$ at a certain *perspective-view plane* in the world space to an omni-image point $X_i(x_i, y_i)$, which can be described as follows [2]:

$$x_i = \frac{f(b^2-c^2)x_p}{(b^2+c^2)z_p - 2bc\sqrt{x_p^2+y_p^2+z_p^2}}, \qquad y_i = \frac{f(b^2-c^2)y_p}{(b^2+c^2)z_p - 2bc\sqrt{x_p^2+y_p^2+z_p^2}}, \tag{1}$$

where $c = \sqrt{a^2+b^2}$ and a, b, and f are the parameters related to the design of the hypercatadioptric camera with f being the camera's focal length. Here, a perspective-view plane is said to be a planar region in the world space on which the above-mentioned projection is conducted. The coordinates (x_i, y_i) of an omni-image point are described with physical units, like millimeter, of the sensor (CCD) plane. Eq. (1) is a one-to-one mapping, which may be denoted as a function h. Then, Eq. (1) can be represented by a simple form as $X_i = h(X_p)$. So after scanning all the points in the perspective-view plane to get the corresponding points in the omni-image, there will be no unfilled pixel in the unwarped image, and the image construction work is completed.

But for a non-SVP designed hypercatadioptric camera, there have no direct one-to-one mapping equation $X_i = h(X_p)$ from $X_p(x_p, y_p, z_p)$ to $X_i(x_i, y_i)$. In [6], the derived equations are divided into two parts. The first part relates a world point $X_p(x_p, y_p, z_p)$ and an unit vector $W_u(w_x, w_y, w_z)$ on the mirror surface with a form of $X_p = g(W_u)$. The second part relates the unit vector $W_u(w_x, w_y, w_z)$ and an omni-image point $X_i(x_i, y_i)$ with an equation $W_u = f(X_i)$. So, the relationship between X_p and X_i is $X_p = g(f(X_i))$. The inverse form $X_i = f^1(g^{-1}(X_p))$ is too complicated to obtain, so it is impossible to conduct the same unwarping task as in the SVP case. In this study, we will solve this unwarping problem using the back-projection form $X_p = g(f(X_i))$ and interpolate the unfilled pixels in the unwarped image by the proposed interpolation method.

2.1 Image unwarping

The task of image unwarping includes the following major steps:

1. calibrating the hypercatadioptric camera;
2. defining a view scope for an unwarped image;
3. creating the unwarped image.

The techniques proposed in this study for these steps are explained in detail in the following.

(1) Calibrating the hypercatadioptric camera

Before using the set of equations derived in [6], the hypercatadioptric camera should be calibrated to determine the pose of the camera with respect to the mirror. The calibrated parameters of the pose include a 3×3 rotation matrix R and a 3×1 translation vector T.

The coordinates of a point $I(u, v)$ on the sensor (CCD) plane in the camera coordinate system with focal length f may be represented as (u, v, f). So, the coordinates of this point I in the mirror coordinate system are:

$$\begin{bmatrix} u_i \\ v_i \\ z_i \end{bmatrix} = R\begin{bmatrix} u \\ v \\ f \end{bmatrix} + T \cdot \tag{2}$$

(2) Defining a view scope for an unwarped image

To get an unwarped image from an omni-image, we should define a view scope under the mirror coordinate system. For a perspective view, the view scope is a *rectangle* defined on a plane paralleling the z-axis of the mirror coordinate system, which is a previously-mentioned perspective view plane. For a panoramic view, the view scope is a cylinder defined between two z-values with the vertical axis paralleling the z-axis of the mirror coordinate system. Figure 3 shows a view scope for a perspective image, and Figure 4 shows a panoramic case.

The view scope is divided into $m \times n$ units to represent $m \times n$ pixels in an unwarped image (for example, 320×240 for a perspective view, 1000×240 for a panoramic view). The height of the view scope is divided into n units both in the perspective view as well as the panoramic view. The m units divide the width of the rectangle in the perspective view case, and divide the surrounding circular range of 360 degrees in the panoramic view case. The coordinates of each unit of the divided view scope in the mirror coordinate system are pre-computed and saved as (x_{pi}, y_{pi}, z_{pi}), $i = 0, 1, \ldots, m\times n - 1$.

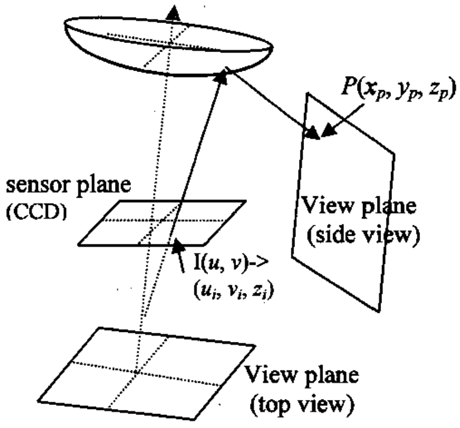

Figure 3: View scope (plane) for viewing the perspective unwarped image.

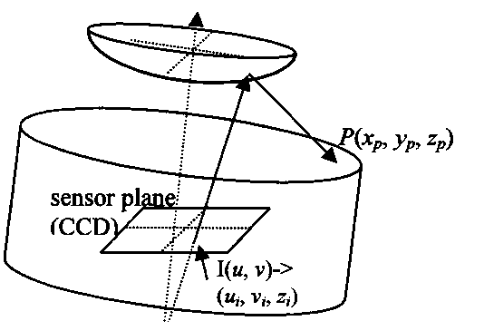

Figure 4: View scope for viewing the panoramic unwarped image.

(3) Creating the unwarped image

As mentioned at the beginning of this section, the back-projection from X_i to X_p is divided into two parts and can be combined as $X_p = g(f(X_i))$. In the first part, $W_u = f(X_i)$ can be pre-computed and saved for later use. In the second part, $X_p = g(W_u)$ is used to set the unwarped image. The details are described as follows.

A. Calculating the relationship $W_u = f(X_i)$

First, the image point $I(u, v)$ in an omni-image is mapped onto an impinging point $M(x_m, y_m, z_m)$ on the mirror by the following equation set [6]:

$$x_m = \frac{-K_5 \pm \sqrt{K_5^2 - K_4 K_6}}{K_4}, \tag{3}$$

$$y_m = y_{cw} - x_{cw}\tan\phi + x_m \tan\phi, \tag{4}$$

$$z_m = -c + b\sqrt{1 + \frac{r_m^2}{a^2}}, \quad r_m^2 = x_m^2 + y_m^2, \tag{5}$$

$$\tan\phi = \frac{v_i - y_{cw}}{u_i - x_{cw}}, \; K_1 = y_{cw} - x_{cw}\tan\phi,$$

$$K_2 = \frac{(z_i - z_{cw})}{u_i - x_{cw}}, \; K_3 = z_{cw} + c - x_{cw}K_2,$$

$$K_4 = b^2(1+\tan^2\phi) - a^2K_2^2, \; K_5 = b^2K_1\tan\phi - a^2K_2K_3,$$

$$K_6 = a^2b^2 + b^2K_1^2 - a^2K_3^2,$$

where (x_{cw}, y_{cw}, z_{cw}) are the coordinates of the camera origin $\boldsymbol{O}$ with respect to the mirror coordinate system, and (u_i, v_i, z_i) are the coordinates of the point $I(u, v)$ on the sensor plane, both can be calculated by Eq. (2). Second, the unit vector $W_u(w_x, w_y, w_z)$ based on the point $\boldsymbol{M}$ can be calculated by the following equation set [6]:

$$w_x = \frac{-(A_mB_m + C_mD_m) \pm \sqrt{(A_mB_m + C_mD_m)^2 - (1 + A_m^2 + C_m^2)(B_m^2 + D_m^2 - 1)}}{(1 + A_m^2 + C_m^2)}, \tag{6}$$

$$w_y = A_m w_x + B_m, \tag{7}$$

$$w_z = C_m w_x + D_m, \tag{8}$$

$$A_m = \frac{\cos\delta K_{m1} + \sin\delta\cos\varphi K_{m3}}{\cos\delta K_{m2} - \sin\delta\sin\varphi K_{m3}}, \quad B_m = \frac{-\cos\rho K_{m3}}{\cos\delta K_{m2} - \sin\delta\sin\varphi K_{m3}},$$

$$C_m = \frac{\sin\delta(\sin\varphi K_{m1} + \cos\varphi K_{m2})}{\cos\delta K_{m2} - \sin\delta\sin\varphi K_{m3}}, \quad D_m = \frac{-\cos\rho K_{m2}}{\cos\delta K_{m2} - \sin\delta\sin\varphi K_{m3}},$$

$$\cos\rho = \frac{(x_{cw} - x_m)\sin\delta\cos\varphi + (y_{cw} - y_m)\sin\delta\sin\varphi - (z_{cw} - z_m)\cos\delta}{\sqrt{(x_{cw} - x_m)^2 + (y_{cw} - y_m)^2 + (z_{cw} - z_m)^2}},$$

$$K_{m1} = (y_m - y_{cw})\cos\delta + (z_m - z_{cw})\sin\delta\sin\varphi,$$

$$K_{m2} = (x_m - x_{cw})\cos\delta + (z_m - z_{cw})\sin\delta\cos\varphi,$$

$$K_{m3} = (x_{cw} - x_m)\sin\delta\sin\varphi - (y_{cw} - y_m)\sin\delta\cos\varphi,$$

$$\sin\delta = \frac{br_m}{\sqrt{a^4 + c^2r_m^2}}, \; \cos\delta = \frac{a\sqrt{a^2 + r_m^2}}{\sqrt{a^4 + c^2r_m^2}}, \; r_m^2 = x_m^2 + y_m^2,$$

$$\varphi = \tan^{-1}\frac{y_m}{x_m},$$

where $c = \sqrt{a^2 + b^2}$ with a and b being two parameters related to the design of the hyperbolic curve of the mirror surface.

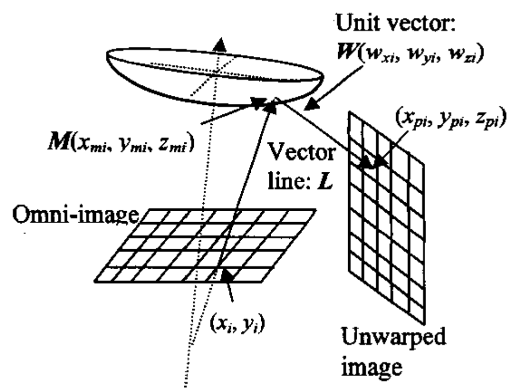

Figure 5: Back-projection of (x_i, y_i).into (x_{pi}, y_{pi}, z_{pi}).
L is described as $t \times \boldsymbol{W} + \boldsymbol{M}$.

B. Unwarping the image

In first part, the relationship $W_u = f(X_i)$ has been computed. We assume that the omni-image has the size of $M_o \times N_o$ pixel^2, So we can store the tri-pairs (x_i, y_i), (x_{mi}, y_{mi}, z_{mi}), and (w_{xi}, w_{yi}, w_{zi}), $i = 1, \ldots, M_o \times N_o - 1$ for further computation, where $X_i(x_i, y_i)$ is represented as (u, v) in part A. In the sequel, the second-part relationship $X_p = g(W_u)$ will be implemented.

Referring to Figure 5, we can describe a point on a vector line $\boldsymbol{L}$ in the mirror coordinate system as follows:

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = t \times \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} + \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix}, \quad (9)$$

where t is a scaling factor. The proposed algorithm to set the value of a pixel in the unwarped image is described briefly as follows:

(a) Setting the value of each of the pixels, denoted as (m_i, n_i), in the $m \times n$ unwarped image to zero.

(b) Calculating the interception point of the vector line $\boldsymbol{L}$ and the view scope.

For each tri-pairs (x_i, y_i), (x_{mi}, y_{mi}, z_{mi}), and (w_{xi}, w_{yi}, w_{zi}), take a point z_{pi} in the view scope and calculate the t value using Eq. (9).

If $t > 0$, calculate $x_p' = t \times w_{xi} + x_{mi}$ and $y_p' = t \times w_{yi} + y_{mi}$. Then, search a point (m_i, n_i) in the unwarped image with real coordinates (x_{pi}, y_{pi}, z_{pi}), which is the nearest point of the calculated point with coordinates (x_p', y_p', z_{pi}). If their distance is smaller than the unit length of the divided view scope, the pixel (m_i, n_i) is marked with a label, say '1'.

(c) Setting the values in the $m \times n$ unwarped image.

If pixel (m_i, n_i) is marked with the label '1', its pixel value is set as the color or grayscale value at point (x_i, y_i) in the omni-image. Otherwise, it is left as an unfilled pixel.

After processing the entire $M_o \times N_o$ points in the omni-image, an $m \times n$ unwarped image is created. Figure 8(a) shows an example of an unwarped image.

2.2 Image interpolation

Image interpolation is a necessary step in image transformation operations, like image scaling, rotation, etc., to interpolate the unfilled pixels between the *expanded pixels* in a resulting higher resolution image. Figure 6 shows a case of two-times expansion, where each expanded pixel (copied from the original pixel) creates 3 unfilled adjacent pixels. An interpolation algorithm should use the information of the expanded pixels close to an undefined pixel to estimate the value to be assigned. There are many previous works on this topic [7, 8]. Two key issues are the major concerns in the design of an interpolation algorithm. One is "how to define the relationship to the nearby pixels? ", and the other one is "what is the information to be used?". Usually, we can model the interpolation of a point x as

$$I(x) = \sum_m c_m h(x_m) \quad (10)$$

where $h(x_m)$ is a relationship function of the m nearby pixels of a pixel x, c_m is the information of one nearby pixel. All the nearby pixels determine the value of the unfilled pixel x as $I(x)$.

In the simplest interpolation algorithm called "pixel reduplication", only one nearby pixel is used, which is the original pixel (with $m = 1$). The relationship function $h(x_m)$ is $h(x_m) = 1$, and the information c_m used is the original pixel value x_m, i. e., $c_m = x_m$. For example, in Figure 6 the values of the 3 unfilled pixels are set to be the same as the expanded one. Of course, some sophisticated interpolation methods like "bilinear", "B-spline" or other modified versions [9, 10] have been proposed to enhance the interpolated quality of the expanded image. There are two important factors in an image expansion problem no matter how complicated the interpolation algorithms are. One is the regular layout of the expanded pixels and the unfilled pixels, and the other is the complete available original image information.

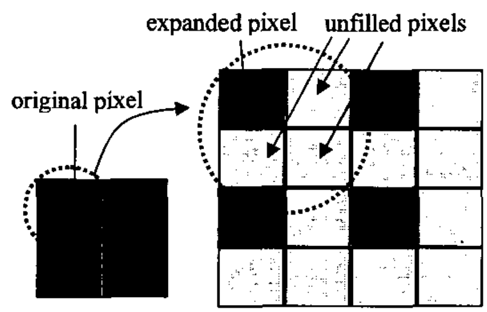

Figure 6: Image expansion.

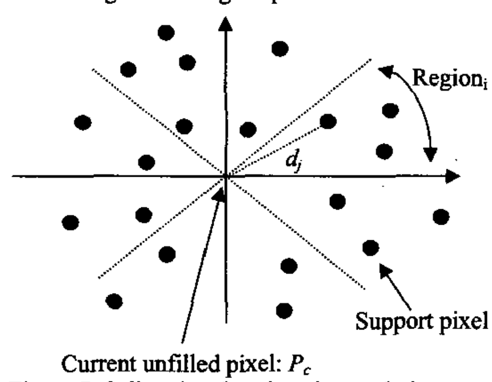

Figure 7: 8 directional regions interpolation. P_c is located at the center of 7×7 matrix.

In this study, the unwarped image from an omni-image suffers from the problem of the irregular distribution of the unfilled pixels that should be interpolated by some methods other than the conventional ones. An omni-image usually is a highly distorted image, in which much geometric information of the original scene is lost. Existing methods in the literature for image expansion do not suit well for our purpose here. So, we propose an *8-directional-regions interpolation* method to interpolate the unwarped image obtained in Section 2.1.

Figure 7 shows the concept of the proposed 8-directional-regions interpolation method. There are 8 directional regions (Region$_i$, i = 0, 1, ..., 7) around an unfilled pixel P_c to be interpolated. In each Region$_i$, there are m_i support pixels P_1, P_2, ..., P_{m_i} (i.e., the expanded pixels in Figure 6) and let the distance from P_j to P_c be denoted as d_j (j = 0, 1, ..., m_i − 1). The set of all the support pixels S_t in the neighborhood of P_c can be described as

$$S_t = \sum_{i=0}^{7} m_i \cdot \quad (11)$$

The region weighting factor w_i for Region$_i$ can be described as

$$w_i = \frac{m_i}{S_t} \cdot \quad (12)$$

Assume that the distance between the current pixel P_c and a support pixel P_j is d_j. Then the distance summation D_i in Region$_i$ can be described as

$$D_i = \sum_{j=0}^{m_i-1} d_j \cdot \quad (13)$$

The weighting factor w_{pj} of each support pixel P_j can be calculated as follows:

$$if (m_i < 2) \ then \ w_{pj} = w_i, \ \ else \ w_{pj} = \frac{D_i - d_j}{(m_i - 1)\times D_i} \times w_i \cdot \quad (14)$$

The pixel value of the current point P_c can be computed by

$$I_c = \sum_{i=0}^{7} I_i ,$$

and the pixel value component in Region$_i$ can be computed by

$$I_i = \sum_{j=0}^{m_i-1} w_{pj} I_{pj} \cdot \quad (15)$$

where I_{pj} is the pixel value of the support pixel P_j.

3 A Visual Surveillance System

We construct a prototype of smart visual surveillance system using a hypercatadioptric camera and a personal computer. The camera is mounted at the ceiling center of a room and the mirror sruface is facing downward to make the FOV of the camera cover the entire room. The kernal algorithms of this system combine the techniques of "object tracking", "view-directed imaging", and "auto-recording" to achieve the goals. The ideas are described as follows.

(1) Pre-building many perspective view planes around the walls of a room.

Twelve view planes, each has 30 degrees of FOV like Figure 3, are set outside the omni-camera and around the walls of a room.

(2) Tracking the object in the current omni-image.

An object tracking method is used to identify the object appearing in the omni-image, and its *view-direction* is calculated. Each view-direction is linked with a view plane to identify which view plane will be used to construct the perspective view image.

(3) Constructing the perspective image.

The perspective view image in the identified view plane is constructed using the proposed method in this paper.

(4) Recording the image sequence.

The image sequence is recorded for further usages from the time when the object appears in a pre-defined time duration.

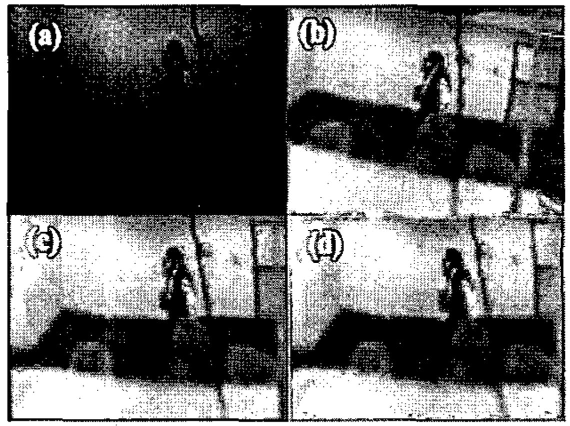

Figure 8: Comparison of constructed perspective side view images with difference methods.
(a) Unwarped image before interpolation using the method proposed in this paper. (b) Unwarped image by the assumption of SVP camera with sensor size 0.8x0.6 mm^2. (c) The interpolation result of (a) by a modified bilinear interpolation method. (d) The interpolation result of (a) by the method proposed in this paper.

Figure 9: Comparison of constructed panoramic images with difference methods.
(a) Unwarped image before interpolation using the method proposed in this paper. (b) Unwarped image by the assumption of SVP camera with sensor size 0.8x0.6 mm^2. (c) The interpolation result of (a) by the method proposed in this paper.

4 Experimental Results

In this section, we present some experimental results to check the image quality yielded by proposed method, as a comparison with those produced by other methods. Figure 8 shows perspective view images constructed by different methods. Figure 8(a) is an unwarped image before interpolation using the method proposed in this paper. Figure 8(b) is the result using Eq. (1) under the assumption that the hypercatadioptric camera used is an SVP-designed one and its sensor size is 0.8×0.6 mm^2 (actually, the sensor size should be 3.2×2.4 mm^2). We can observe some notable problems in this image, for example, the obvious distortion of the geometric shape of the scene and the more fuzzy boundary parts of the image. Figures 8 (c) and (d) are the results constructed by the proposed unwarping method but using different image interpolation methods. Figure 8(d) is the best result in the respect of geometric shape recovery and image quality. Figure 9 shows the results of panoramic images.

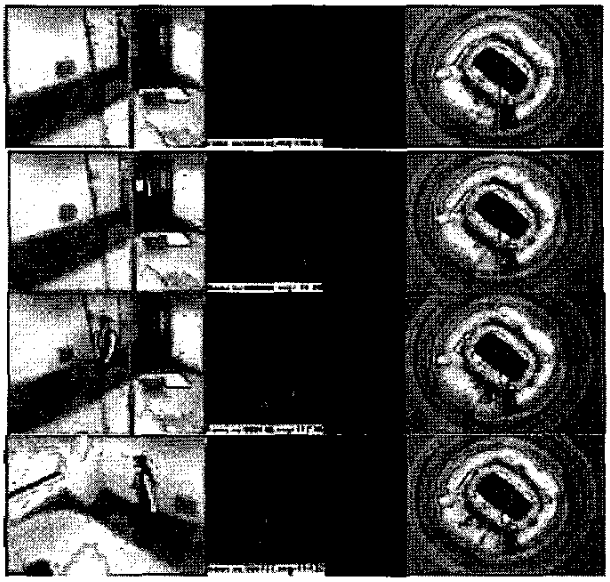

Figure 10: Some snapshots in a video sequence.
Left column: the constructed perspective image.
Middle column: the tracked object.
Right: the omni-image.

Figure 10 shows some snapshots in a video sequence taken by a prototype smart visual surveillance system mentioned in Section 3. The left column shows the constructed perspective image in the detected view-direction. The middle column shows the tracked object in the omni-image. The view-direction is calculated as a rotation angle with respect to the center of the omni-image. The right column shows the current omni-image. When the rotation angle of a tracked object is within the FOV of a view plan, the corresponding perspective image is displayed. One obvious advantage of such a system is that the target is always located in the center part of the perspective image.

5 Conclusions

In this paper, we have proposed a method of image construction from an omni-image taken by a non-SVP designed hypercatadioptric camera. Image construction cannot be simplified by the assumption that the current hypercatadioptric camera is an SVP designed one, Figure 8(b) or Figure 9(b) shows the result by this assumption. So, in this study, we solve the image construction problem by the combination of the techniques of unwarping images with analytic equations and interpolating the unwarped image with the 8-directional-regions method. The results are acceptable for use in a visual surveillance system in the respect of computation time and image quality. For example, in Figure 10, the average cycle time for processing an omni-image is less than 130 mini-seconds, which include the times for image processing and displaying, when a Pentium III computer with a 1.13GHz CPU is used.

References

[1] S. Nayar, and T. Boult, "Omnidirectional VSAM system: PI report," *Proceedings of DARPA Image Understanding Workshop*, pp. 55-61, May 1997.

[2] Y. Onoe, N. Yokoya, K. Yamazawa, and H. Takemura, "Visual Surveillance and Monitoring System Using an Omnidirectional Video Camera," *Proceedings of Fourteenth International Conference on Pattern Recognition*, vol. 1, pp. 588-592, Aug 1998.

[3] C. Geyer and K. Daniilidis, "Paracatadioptric Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 687-695, May 2000.

[4] S. K. Nayar, "Catadioptric Omnidirectional Camera," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 482-488, June 1997.

[5] R. Swaminathan, M. D. Grossberg, and S. K. Nayar, "Caustics of Catadioptric Cameras," *Proceedings of Eighth IEEE International Conference on Computer Vision*, vol. 2, pp. 2-9, 2001.

[6] S. W. Jeng, and W. H. Tsai, "Precise Image Unwarping of Omnidirectional Cameras with Hyperbolic-Shaped Mirrors,", in *Proceedings of 16th IPPR Conference on Computer Vision, Graphics and Image Processing*, Kinmen, R. O. C , pp. 414-422, August 17~19, 2003.

[7] T. M. Lillesand, and R. W. Kiefer, "Remote Sensing and Image Interpretation," 1994, 3rd Ed. John Wiley and Sons, Inc. 750pp.

[8] L. Polidori,, and J. Chorowicz, "Comparison of Bilinear and Brownian Interpolation for Digital Elevation Models," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 2, no 48, pp. 18-23, 1993.

[9] S. W. Lee and J. K. Paik, "Image Interpolation Using Adaptive Fast B-spline Filtering," in *Proceedings of IEEE International Conference on Acoustics, Speech, Signal Processing*, vol. 5, pp. 177-180, 1993.

[10] J. Allebach and P. W. Wong, "Edge-Directed Interpolation," in *Proceedings of IEEE International Conference of Image Processing*, vol. 3, pp. 707-710, 1996.