Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 28 (2007) 1926–1936
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Improving quality of unwarped omni-images with irregularly-distributed unfilled pixels by a new edge-preserving interpolation technique

Sheng-Wen Jeng [a,*], Wen-Hsiang Tsai [a,b,1]

[a] Department of Computer Science, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu, Taiwan 30010, Taiwan
[b] Department of Computer Science and Information Engineering, Asia University, Liufeng Road, Wufeng, Taichung, Taiwan 41354, Taiwan

Received 27 March 2006; received in revised form 6 March 2007
Available online 24 May 2007

Communicated by R. Davies

Abstract

A new method called "edge-preserving 8-directional two-layered weighting interpolation" is proposed for interpolating unfilled pixels in a perspective-view or panoramic image resulting from unwarping an omni-image taken by a non-single-view-point hypercatadioptric camera. This method can solve the problem of edge preserving in interpolating the input image which has many irregularly-distributed unfilled pixels. Three concepts of weighting have been considered in the proposed edge-preserving interpolation process: (1) inverse-distance weighting; (2) pixel-count weighting; and (3) region-wise weighting. Good experimental results showing the effectiveness of edge preserving and its superiority to other methods are also included.

Keywords: Omni-image; Hypercatadioptric camera; Edge-preserving; Irregular; Interpolation

1. Introduction

A catadioptric *omni-directional camera* (or simply, *omni-camera*) (Nayar, 1997a,b), which is a combination of a reflective mirror and a conventional camera, captures the incoming light to form an *omni-directional image* (or simply, *omni-image*). The mirror surface may be of various shapes, like conic, parabolic or hyperbolic, etc. If all the reflected light rays pass through a common point, the camera is said additionally to be of the *single-view-point* (SVP) type (Baker and Nayar, 1999); otherwise, of the *non-single-view-point* (non-SVP) type. On the other hand, a *hypercatadioptric camera* is a type of catadioptric omni-camera with a hyperbolic-shaped mirror (Yamazawa et al., 1993a,b). Fig. 1 shows the structure of an SVP hypercatadioptric camera, where the focus point of the camera is located at the outer focus point O_c of the hyperbolic curve of the mirror surface. That is, in this perfectly aligned structure, all the incoming light rays pass through O_c. If the structure is instead mis-aligned, the incoming light rays will not pass through O_c, but form a so-called "caustic" surface (Swaminathan et al., 2001), and the camera becomes a non-SVP one.

It is difficult to unwarp an omni-image into a normal-view one, and the degree of difficulty depends on the type of omni-camera and the alignment of the camera structure. SVP omni-images are generally easier to handle (Peri and Nayar, 1997; Onoe et al., 1998; Ying and Hu, 2004) than non-SVP ones. If a non-SVP omni-image is treated as an SVP one and unwarped accordingly, the resulting perspective-view image, called *unwarped image* in the sequel, will

* Corresponding author. Tel.: +886 6 3847167; fax: +886 6 3847297.
E-mail addresses: sunny@itri.org.tw (S.-W. Jeng), whtsai@cis.nctu.edu.tw (W.-H. Tsai).
[1] Tel.: +886 3 5728368.

0167-8655/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2007.03.021

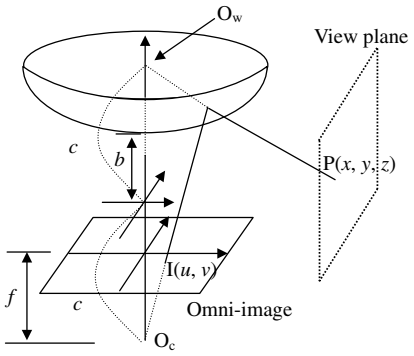

Fig. 1. An SVP hypercatadioptric camera where O_w is one focus of the hyperbolic curve taken to be the origin of the world coordinate system, and O_c is the optical lens center as well as the other focus of the hyperbolic curve.

suffer a serious geometric distortion (Mashita et al., 2006), especially when the structural misalignment is *large* in some camera designs aiming to extending the field of view (FOV) of the camera. Fig. 2b shows this case. So, in the case of *non-SVP image unwarping*, we need another way to do the unwarping job.

More specifically, unwarping an omni-image, which is taken with an SVP hypercatadioptric camera, into a perspective version is a process of *forward projection* from a point $\boldsymbol{X}_p = (x, y, z)$ on a certain perspective-view plane in the world space to an omni-image pixel $\boldsymbol{X}_i = (u, v)$, which can be described by $\boldsymbol{X}_i = h(\boldsymbol{X}_p)$ with h being a *one-to-one* mapping function from the world space to the omni-image plane (Onoe et al., 1998; Jeng and Tsai, 2004). Eq. (1) below is a description of this mapping from (x, y, z) to (u, v):

$$u = \frac{f(b^2 - c^2)x}{(b^2 + c^2)z - 2bc\sqrt{x^2 + y^2 + z^2}},$$
$$v = \frac{f(b^2 - c^2)y}{(b^2 + c^2)z - 2bc\sqrt{x^2 + y^2 + z^2}}, \quad (1)$$

where f is the focal length of the camera lens, and a, b and c are the parameters of the hyperbolic curve of the mirror surface described by the following equations:

$$z = -c + b\sqrt{1 + \frac{r^2}{a^2}}, \quad r^2 = x^2 + y^2, \quad c = \sqrt{a^2 + b^2}. \quad (2)$$

Consequently, after scanning all the points (x, y, z) in the perspective-view plane in the world space to get the coordinates of the corresponding pixel (u, v) in the omni-image in the unwarping process using Eq. (1), there will be *no unfilled pixel* in the resulting image because of the one-to-one mapping property.

However, for a non-SVP hypercatadioptric camera, there exists no *direct* one-to-one mapping $\boldsymbol{X}_i = h(\boldsymbol{X}_p)$ from $\boldsymbol{X}_p$ to $\boldsymbol{X}_i$, and the relationship between $\boldsymbol{X}_p$ and $\boldsymbol{X}_i$ instead is $\boldsymbol{X}_p = g(f(\boldsymbol{X}_i))$ with f being a mapping from the omni-image plane to the mirror surface and g another mapping from the mirror surface to the world space (Jeng and Tsai, 2004). The inverse forms g^{-1} and f^{-1} of the mappings are too complicated to obtain such that the direct mapping $\boldsymbol{X}_i = h(\boldsymbol{X}_p) = f^{-1}(g^{-1}(\boldsymbol{X}_p))$ is unavailable. Therefore, it is impossible to conduct the same type of unwarping task

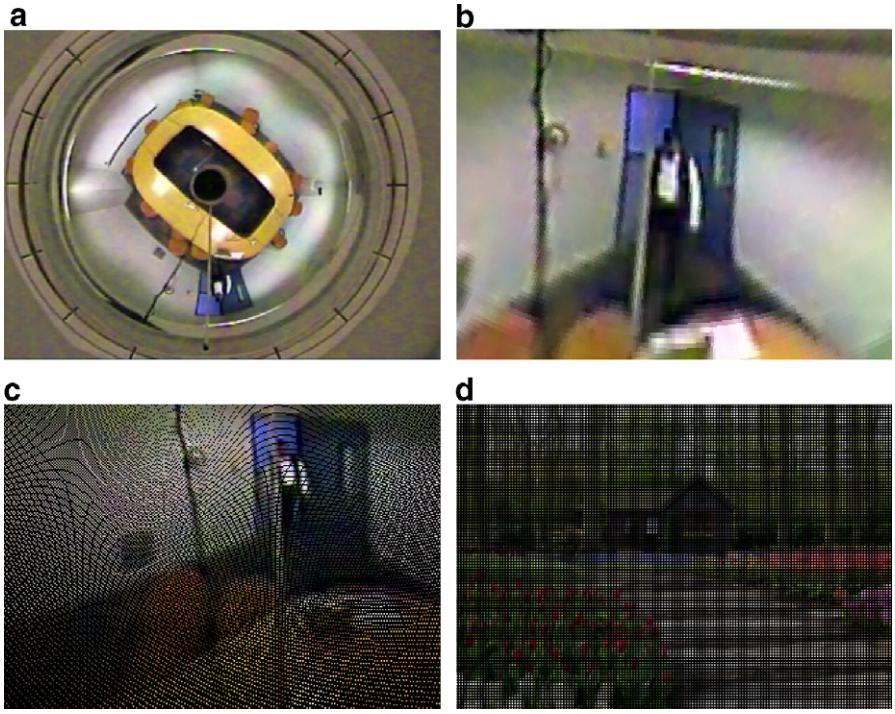

Fig. 2. Examples of images for illustration. (a) An omni-image taken by a non-SVP hypercatadioptric camera. (b) An unwarped perspective-view image when (a) is treated as an SVP camera. (c) Raw image resulting from unwarping the omni-image of (a) using a back projection method proposed by Jeng and Tsai (2004). (d) Raw enlarged image of a scene image with regularly-tessellated unfilled pixels before interpolation.

as in the SVP case. Jeng and Tsai (2004) solved this unwarping problem by a *back projection* method using the original mapping $\boldsymbol{X}_p = g(f(\boldsymbol{X}_i))$, yielding in the unwarped image many *irregularly-distributed unfilled pixels* to which no omni-image data have been mapped. Fig. 2c shows an example of the unwarping result. The unfilled pixels should be filled by some interpolation method.

Conventional interpolation methods (Jain, 1989; Atwood and Davis, 1989) deal mainly with *regularly-tessellated unfilled pixels*, resulting from conventional image transformations like image scaling (see Fig. 2d for an example), and so are unsuitable for handling the *irregularly-distributed* unfilled pixels in the unwarped image. In (Li and Orchard, 2001), Li and Orchard proposed a new edge-directed interpolation by using the geometric duality between the low-resolution covariance and the high-resolution one of the image, but it is applicable only to images with *regularly-distributed* unfilled pixels created by image expansion. In our case here, the property of geometric duality does not exist, so the method is inapplicable. Jeng and Tsai (2004), in addition to the back projection method, have proposed a so-called 8-*directional-regions interpolation technique* to do this work. However, there is a drawback in their method, that is, *edge blurring* caused by the averaging operation in the method will occur in the interpolation result.

In this study, we propose a new method called "edge-preserving 8-directional two-layered weighting interpolation" to conduct the desired interpolation work, which can preserve local edges in images to avoid the edge-blurring effect. The method is thus more suitable to the unwarped image which has many irregularly-distributed unfilled pixels. An edge-line detection scheme is proposed first for deciding if an unfilled pixel lies on an edge line. If the unfilled pixel is found to be on an edge line, then the pixel value is assigned by a scheme of inverse-distance weighting of two filled pixels on the edge line. Otherwise, a *two-layered weighting* interpolation scheme is applied, which considers *three* concepts of weighting, namely, inverse-distance weighting, pixel-count weighting, and region-wise weighting. This scheme uses the ratio of filled-pixel counts of each 8-directional sub-region to the total filled-pixel counts in all sub-regions as the first-layer weighting factor, and then uses the inverse distances between the filled pixels in each sub-region and the currently-processed unfilled pixel as the second-layer weighting factors to complete an even pixel-value contribution effect in the interpolation. The quality of the interpolation result is greatly improved in the aspect of edge preserving. It is noted here that this type of interpolation work for processing unwarped non-SVP omni-images, to the knowledge of the authors, has not been investigated before. Although the idea of edge preserving can be found in (Ramponi and Carrato, 2001), which investigated irregular sampling for image coding, here we utilize the edge information in a different way to process the more complicated unwarped image *with irregularly-distributed pixels at unknown positions*. Also, the weighting scheme in utilizing neighboring pixel values adopted in (Jeng and Tsai, 2004) is modified to improve the interpolation accuracy.

In the remainder of this paper, we describe the proposed method in Section 2, show the experimental results in Section 3, and make conclusions finally in Section 4.

2. Image interpolation by edge preserving and pixel value weighting

The proposed method is an *edge-preserving* 8-*directional two-layered weighting interpolation technique* which, as mentioned previously, is based on three types of weighting for interpolation: (1) inverse-distance weighting; (2) pixel-count weighting; and (3) region-wise weighting. We first describe in Section 2.1 the core of the proposed technique involving *no edge preserving*, and then the proposed *edge-preserving version* of the proposed interpolation technique in Section 2.2.

2.1. Non-edge-preserving version of proposed method

As illustrated in Fig. 3, the non-edge-preserving version of the proposed technique is based on an $n \times n$ region R (e.g., a 7×7 region) centered at an unfilled pixel P_{u} whose value is to be interpolated. Region R is divided into eight sub-regions $S_1, S_2, \ldots, S_8$. Filled pixels in each sub-region, called *support pixels* hereafter, are used in calculating the weights for interpolation.

Let the support pixels in sub-region S_i $(i = 1, 2, \ldots, 8)$ be denoted as $P_{i1}, P_{i2}, \ldots, P_{im_i}$ and their respective pixel values denoted as $I_{i1}, I_{i2}, \ldots, I_{im_i}$, where m_i is the number of such pixels in S_i. Let the distance from a support pixel P_{ij} to P_{u} be denoted as d_{ij} where $j = 1, 2, \ldots, m_i$. Let the number of all the support pixels in the neighborhood R of P_{u} be denoted as M, which is equal to

$$M = \sum_{i=1}^{8} m_i. \tag{3}$$

The *region weighting factor* s_i for sub-region S_i based on the pixel numbers is defined as

$$s_i = \frac{m_i}{M}. \tag{4}$$

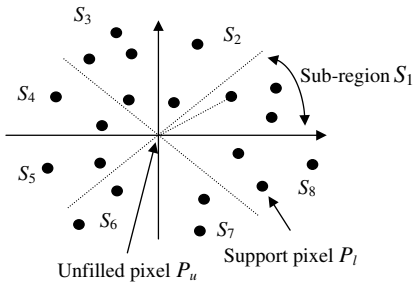

Fig. 3. Illustration of proposed interpolation.

And the *distance weighting factor* d_i based on *inverse pixel distances* for sub-region S_i is defined as

$$d_i = \sum_{j=1}^{m_i} (1/d_{ij}). \tag{5}$$

Then, the weight factor w_{jj} of support pixel P_{ij} for use in the interpolation is calculated as

$$w_{ij} = \frac{(1/d_{ij})}{d_i} \times s_i. \tag{6}$$

Now, the proposed two-layered weighted interpolation can be described as follows. First, we compute the *contribution* I_i of sub-region S_i to the desired interpolation value I_u of P_u as

$$I_i = \sum_{j=1}^{m_i} w_{ij} I_{ij}, \tag{7}$$

where I_{ij}, as described previously, is the pixel value of support pixel P_{ij}. And then, the desired pixel value of P_u can be computed by

$$I_u = \sum_{i=1}^{8} I_i. \tag{8}$$

Note that in case there exists no support pixel in one or more of the eight sub-regions, the weights of such sub-regions will be ignored automatically according to the above process, and only those of the other sub-regions with existing support pixels will be used in calculating the value I_u of the unfilled pixel P_u. This can be seen from Eq. (4) where s_i will be equal to 0 when $m_i = 0$. In the extreme case that no support pixel can be found in all the sub-regions, then we adopt the approach of enlarging the $n \times n$ region R, e.g., from 7×7 to 9×9, and so on, until at least one support pixel is found in the enlarged R.

A merit of the above proposed approach of *two-layered interpolation* is that the contribution of a sub-region S_i, as can be figured out from the above formula, can be controlled by the corresponding region weighting factor s_i which is computed from the number of support pixels in S_i. This will limit the magnitude of the contribution of the sub-region, compared with the traditional scheme of one-layered inverse-distance weighting which will cause uneven or dominant contribution of certain pixels at very short distances to the currently-processed unfilled pixel P_u. Another merit is that more contribution will be yielded by a sub-region with more support pixels, as can be seen again from Eq. (4) for computing s_i. This is reasonable because more information is conveyed by more pixels and such information should be utilized properly by giving larger weights to them. These two merits are advantageous to the work of unwarping the *irregularly-distributed and spatially-sparse pixels* in the unwarped image which we are dealing with, as proved by our experimental results described later.

2.2. Edge-preserving version of proposed technique

In the above non-edge-preserving version of the proposed technique, the weight of a support pixel basically is determined by its distance to the currently-processed unfilled pixel P_u. All the support pixels are used to accomplish the interpolation work. In the edge-preserving version of the proposed technique, each support pixel instead will be classified first as an *edge pixel* or not, using the edge pixel information found in the original omni-image. And the spatial relationship between P_u and the edge pixels are analyzed to determine which support pixels should be used to conduct the interpolation. The essence of the idea behind the proposed technique is to ignore non-edge pixel contribution to the interpolation if an edge is found to go through P_u, thus reducing the blurring effect in the unwarped image.

Now the issue is how to find the edge pixels for use in the above process. Although edge pixels can be easily found by traditional edge detection methods in the original omni-image which is rectangularly-tessellated in nature, what we want is their corresponding pixels in the unwarped image. Since such pixels in the unwarped image are irregularly and sometimes even sparsely distributed, traditional edge detection methods are inapplicable. To solve this problem, we compute first an *edge-value image* from the original omni-image, and then unwarp it into a second image and threshold the resulting image values to get an *unwarped edge map*. Finally, we detect edges in the unwarped edge map by a way of checking edge lines formed by triple pixels. The details are described in the following. Let the original omni-image be denoted as I_o.

2.2.1. Creation of unwarped edge map

First, we compute an edge-value image I_e by the Sobel operator with I_o as input and normalize the edge values in I_e into the range of 0.0–1.0. Fig. 4a shows an input omni-image I_o and Fig. 4c shows the resulting I_e computed from Fig. 4a.

Second, the original omni-image and the edge-value image are unwarped with the non-SVP forward-projection method proposed in (Jeng and Tsai, 2004) to get a raw unwarped image and an unwarped edge-value image, respectively. The unwarped edge-value image is then thresholded into an unwarped edge map by a threshold value e_t. Fig. 4b and d show the results, respectively. There are many unfilled pixels in the raw unwarped image, whose values are what we want to interpolate in the sequel. And the unwarped edge map will be used to assist this task.

2.2.2. Performing local edge analysis to detect support pixels on edges

An unfilled pixel P_u is regarded to be *on an edge line* if P_u *and* two support pixels P_k and P_l forms *approximately* a straight line, where P_k and P_l lie respectively in two sub-regions S_i and S_j which are opposite in directions (e.g., S_1 and S_5, S_2 and S_6, and so on). To detect such

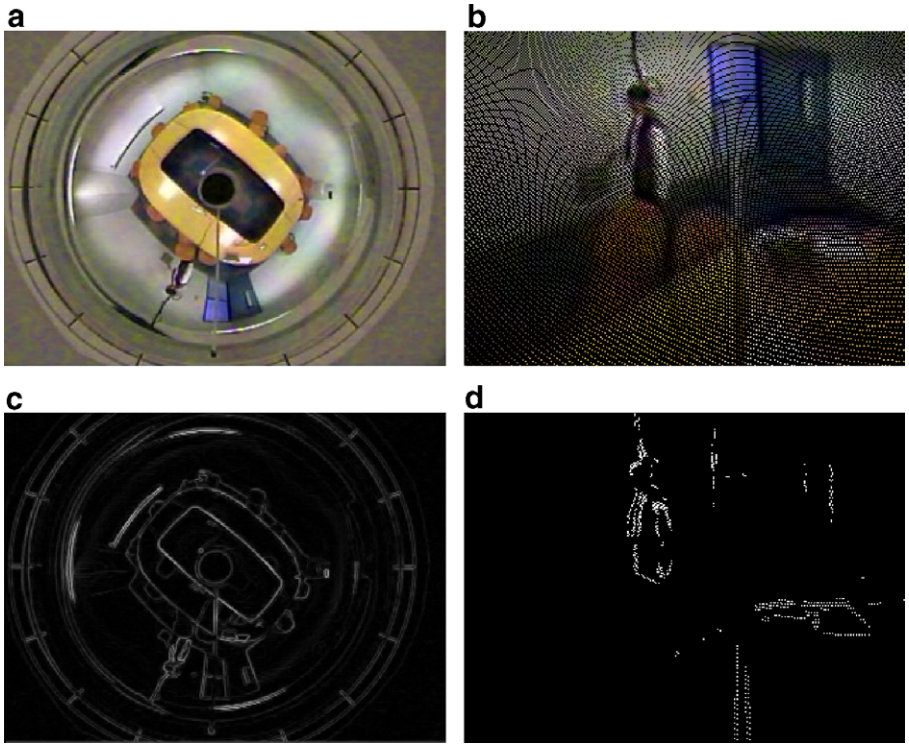

Fig. 4. Creation of unwarped edge map from original omni-image. (a) Original omni-image. (b) Raw unwarped image. (c) Edge-value image of (a). (d) Unwarped edge map (with $e_t = 0.3$).

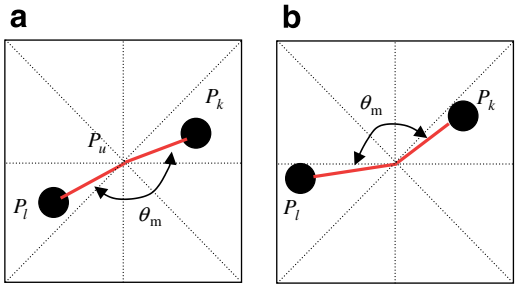

Fig. 5. Illustration of proposed interpolation. (a) Edge detected (θ_m close to 180°). (b) No edge detected (θ_m not close to 180°).

on-edge support pixels, every possible pair of oppositely-directed sub-regions S_i and S_j and every pair of support pixels P_k and P_l located, respectively in them are selected and analyzed in the following way:

(a) calculate the angle θ_m formed by the two vectors $\boldsymbol{P_u P_k}$ and $\boldsymbol{P_u P_l}$;
(b) if θ_m is close to 180° (determined by checking if it is larger than a threshold θ_t), then regard P_u to be on an edge line and record the pair of support pixels P_k and P_l as a *candidate pair*. Fig. 5 illustrates this idea.

2.2.3. Interpolation using on-edge support pixels

We then take from the set of all candidate pairs the one with support pixels P_a and P_b which forms an angle θ_m *closest* to 180°, and use their distances d_a and d_b to P_u to complete the desired interpolation for the pixel value I_u of P_u in the following way:

$$I_u = [(1/d_a) \times I_a + (1/d_b) \times I_b]/[(1/d_a) + (1/d_b)]. \quad (9)$$

That is, we compute the pixel value I_u by an interpolation from those of P_a and P_b weighted inversely by their respective distances d_a and d_b to P_u.

2.2.4. Interpolation by non-edge support pixels

If no candidate pair is found in the above step, then we conduct interpolation of the value of I_u for pixel P_u by the non-edge-preserving version of the proposed technique described in Section 2.1.

3. Experimental results

In the field of image interpolation, most existing objective metrics of image quality are unsuitable to evaluate the quality of the results of the applied methods, as mentioned in (Li and Orchard, 2001). Subjective evaluation via human eyes is more practical, and the improvements of the proposed method may be easily observed in the figures of the following experimental results.

In our implementation of the proposed method, there are two threshold parameters, the edge value threshold e_t and the angle span threshold θ_t, which can be adjusted for defining edge points and checking the on-edge condition, respectively, as described previously. The value of e_t

is set between 0 and 1 representing normalized edge intensity values. The value of θ_t is set between 0° and 180° representing the angle spanned by two support pixels around an unfilled pixel. Fig. 6 shows a comparison of an experimental result of the proposed method by setting $e_t = 0.35$, $\theta_t = 120°$ with that of a conventional method. Fig. 6a shows the original omni-image. Fig. 6b shows the raw unwarped image which contains many unfilled pixels to be interpolated. Fig. 6c is the result using the traditional 4-nearest-neighbor (4NN) interpolation method with 7×7 window size. Fig. 6d shows the unwarped edge map in which the white points are the edge pixels obtained from thresholding the unwarped edge-value image (called *unwarped edge pixels*) and the blue points are unfilled pixels detected to be on edge lines in the unwarped edge map (called *on-edge pixels*). Fig. 6e is the result using the edge-preserving version of the proposed method, and Fig. 6f is the result using the non-edge-preserving version of the proposed method. The overall image quality improvement can be found obvious by comparing the results of Fig. 6e and f with that of Fig. 6c. Furthermore, we can see the edge quality improvement in Fig. 6e near the locations of the edge pixels in Fig. 6d. The processing times for Fig. 6c, e and f of the size 320×240 using a PC with an Intel P4 3 GHz CPU and a 1 GB RAM are 90 ms, 218 ms, and 94 ms, respectively. The need of longer processing time for Fig. 6e than those for Fig. 6c and f is owing to the time-consuming step of computing the edge map of Fig. 6d. Note that the computations of Fig. 6c and f do not involve the edge map, but the cost of saving time in this aspect is the blurring effect at edges.

Fig. 7 shows the results of applying the proposed method to a raw unwarped image, Fig. 6b, using different values for the threshold e_t and a constant value 120° for the threshold θ_t. Lower values of e_t will yield more edge points and so more on-edge unfilled pixels, as can be seen

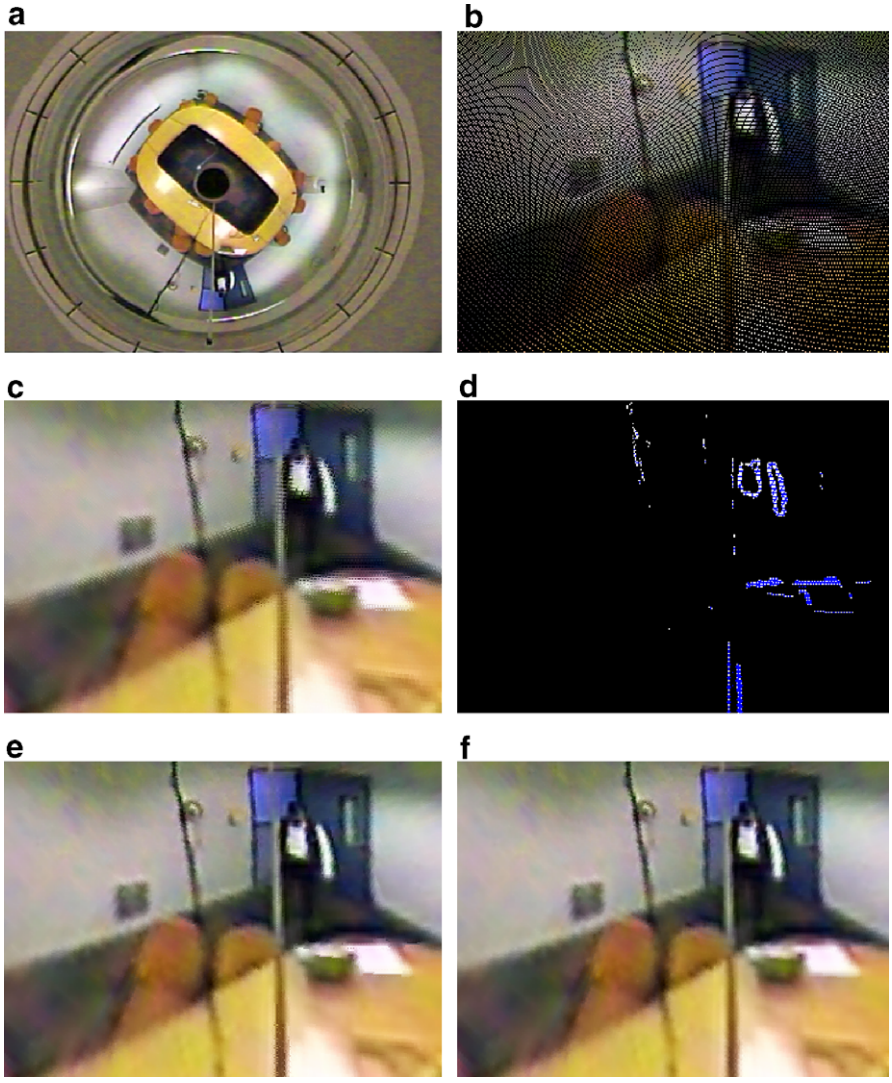

Fig. 6. Comparison of results of different methods. (a) Original image. (b) Raw unwarped image with unfilled pixels. (c) Interpolation result using 4 NN method (processing time = 90 ms). (d) Detected edge pixels (white) and on-edge pixels (blue). (e) Result of proposed method with edge preserving (processing time = 218 ms). (f) Result of proposed method without edge preserving (processing time = 94 ms). (For interpretation of the figure in colour, the reader is referred to the web version of this article.)

in Fig. 7a and b, causing improper blurring effects near the edge locations. Higher values of e_t will reduce the number of resulting edge points, yielding a result of Fig. 7f which is almost the same as that obtained from using the non-edge-preserving version of the proposed method. After many experimental trials, we suggest the use of the edge threshold value range of $0.2 < e_t < 0.4$ for use in unwarping non-SVP omni-images.

Fig. 8 shows the results of using different values for θ_t, ranging from 120° to 180°, and a constant value 0.35 for e_t, with the raw unwarped image of Fig. 6b as input. It is difficult to see visual differences among the global image qualities of these results. But we can inspect the detailed distribution of the on-edge pixels by two enlarging portions of the images of Fig. 8a and e, as shown in Fig. 9a and b, respectively.

From Fig. 9b, we can see that a very *tight* "on-edge" condition check, using the extreme value of $\theta_t = 180°$, will result in losing some real on-edge pixels because at the "digital pixel" level, a straight edge-line formed by two different pixels does not always pass precisely the middle pixel of the currently-processed window. So, we should *relax* the "on-edge" checking condition to allow more reasonable pixels (for example, the point P_c in Fig. 9b) to be accepted as on-edge pixels, as Fig. 9a shows.

More specifically, referring to Fig. 10, we calculate the span angle θ_m of the unfilled pixel P_c by the following equation according to trigonometry:

$$\theta_m = \cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2 \times a \times b}\right). \tag{10}$$

By substituting proper values of a, b, and c in Fig. 10, namely, $a = \sqrt{2}$, $b = 1$, and $c = \sqrt{5}$, into Eq. (10), we get $\theta_m = \cos^{-1}(\frac{2+1-5}{2\times\sqrt{2}\times 1}) = \cos^{-1}(-\sqrt{0.5}) = 2.356$ which is equal to 135°. So, the range of the span angle θ_m is taken to be

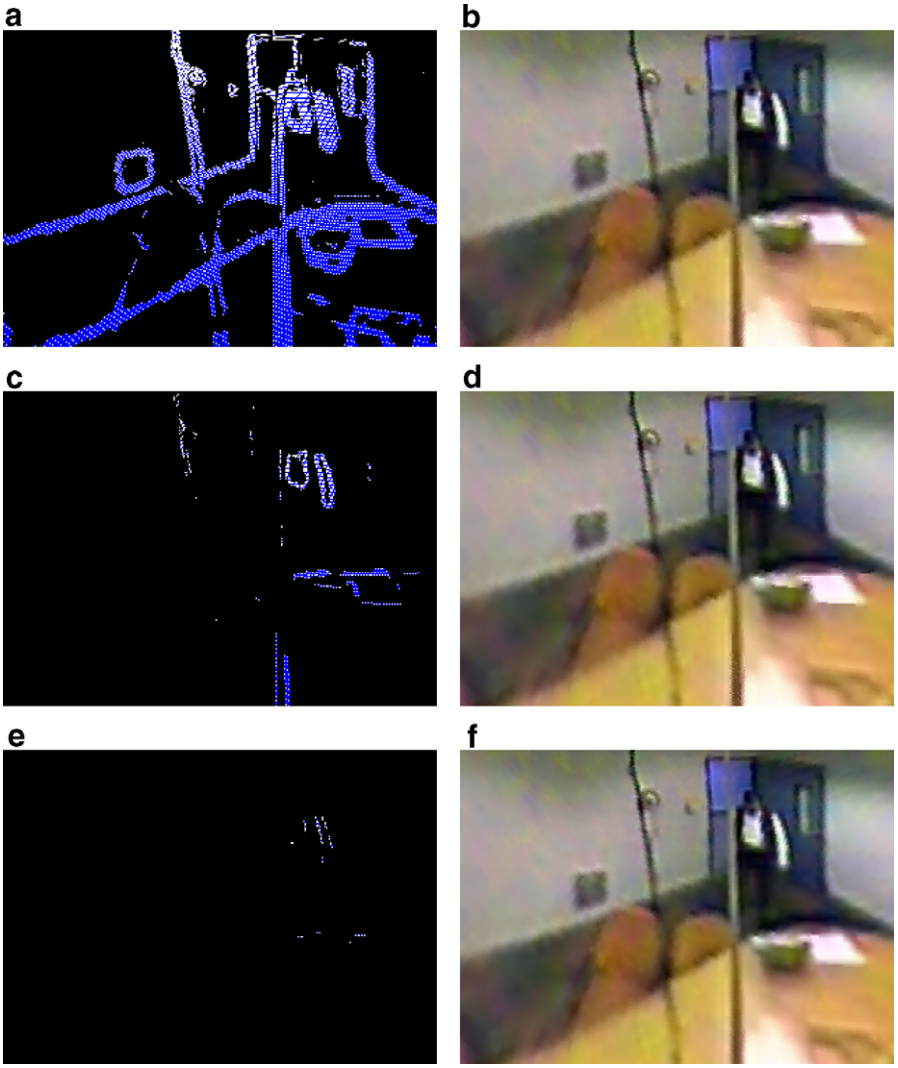

Fig. 7. Results using different edge threshold values at constant angle threshold ($\theta_t = 120°$). (a) Unwarped edge pixels (white) and detected on-edge pixels (blue) with $e_t = 0.1$. (b) Resulting image of proposed method with $e_t = 0.1$. (c) Unwarped edge pixels and detected on-edge pixels with $e_t = 0.35$. (d) Resulting image of proposed method with $e_t = 0.35$. (e) Unwarped edge pixels and detected on-edge pixels with $e_t = 0.6$. (f) Resulting image of proposed method with $e_t = 0.6$. (For interpretation of the figure in colour, the reader is referred to the web version of this article.)

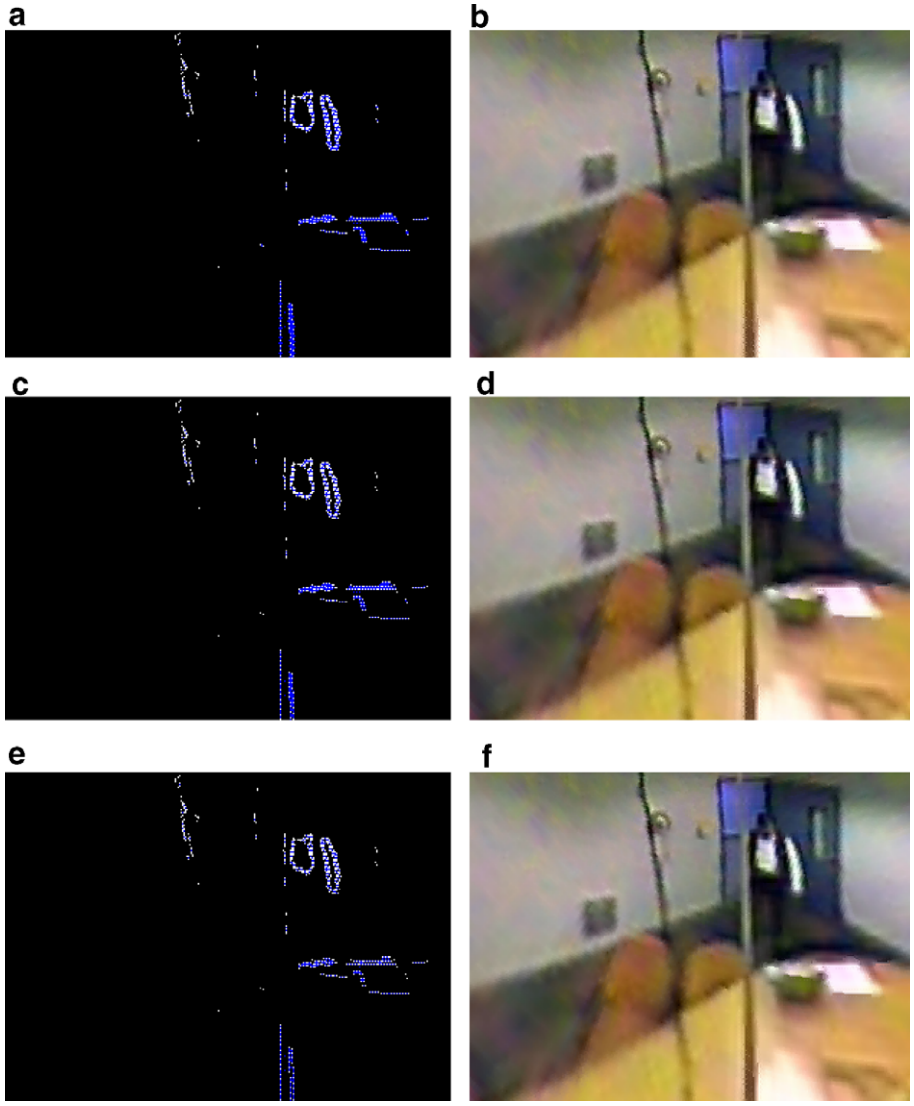

Fig. 8. Results using different angle threshold values at constant edge threshold ($e_t = 0.35$). (a) Unwarped edge pixels (white) and detected on-edge pixels (blue) with $\theta_t = 120°$. (b) Resulting image of proposed method with $\theta_t = 120°$. (c) Unwarped edge pixels and detected on-edge pixels with $\theta_t = 150°$. (d) Resulting image of proposed method with $\theta_t = 150°$. (e) Unwarped edge pixels and detected on-edge pixels with $\theta_t = 180°$. (f) Resulting image of proposed method with $\theta_t = 180°$. (For interpretation of the figure in colour, the reader is referred to the web version of this article.)

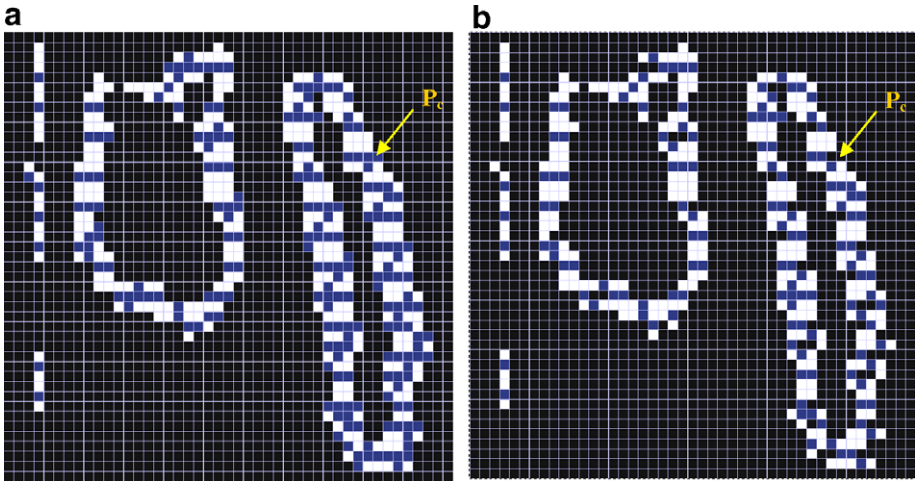

Fig. 9. Detailed edge maps obtained by using different angle thresholds and an identical edge threshold. (a) Angle threshold $\theta_t = 120°$. (b) Angle threshold $\theta_t = 180°$. Note that P_c is an on-edge pixel in (a) (marked by blue color), but is not in (b) (appearing as black). (For interpretation of the figure in colour, the reader is referred to the web version of this article.)

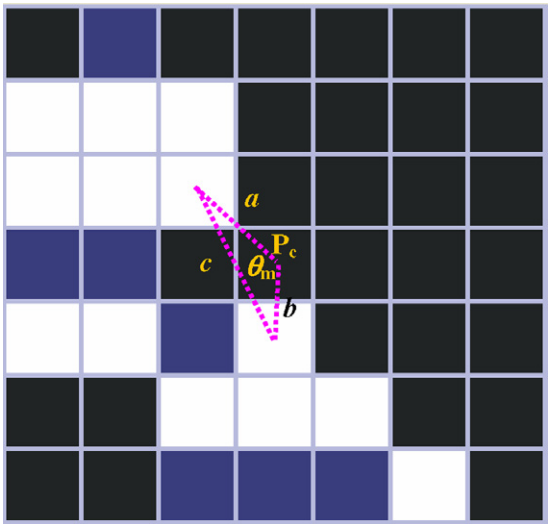

Fig. 10. Detailed calculation of the span angle of an unfilled pixel.

larger than 120° and smaller than 180°. In this sense, P_c is an on-edge pixel in Fig. 9a because the threshold θ_t is taken to be 120°, and it is not in Fig. 9b because of the tight threshold value $\theta_t = 180°$.

Furthermore, in Fig. 11, we try the proposed method using different window sizes to a raw unwarped image, Fig. 6b. The processing time increases with the window size, as expected. When the window size is reduced to 5×5, some unfilled pixels become to have no support pixel and leave holes (black dots) in the image (for example, see the region in the lower part of Fig. 11a enclosed by the dotted circle). On the contrary, when the window size is larger than 7×7, the image quality is almost kept the same. So, it is not necessary to use window sizes larger than 7×7.

Finally, Fig. 12 shows a comparison of the results of different interpolation methods for the case of interpolating *regularly*-distributed unfilled pixels in images. Fig. 12a is an image produced from one by duplicating the pixel lines without filling the duplicated pixels. And Fig. 12b–d are the interpolation results of three methods including the proposed one. What we want to show here is that the proposed method is also applicable to the conventional image expansion problem, yielding again results of better quality than those of other methods.

4. Conclusions

We have proposed in this paper a method of "edge-preserving 8-directional two-layered weighting interpolation" for processing an unwarped image with irregularly-distributed unfilled pixels which is the result of unwarping an omni-image taken from a non-SVP hypercatadioptric camera. In addition to possessing the edge-preserving capability, the method takes into consideration three concepts of weighting for the involved interpolation scheme: (1) inverse-distance weighting; (2) pixel-count weighting; and

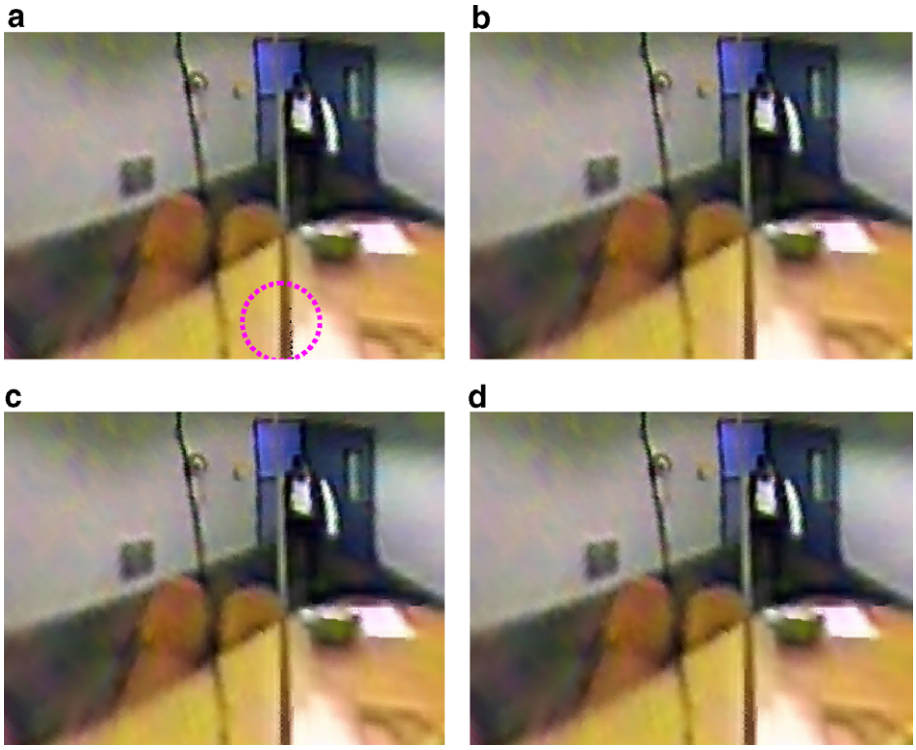

Fig. 11. Interpolation results with different window sizes. The processing time increases with the window size. (a) Window size 5×5 (processing time = 172 ms). (b) Window size 7×7 (processing time = 234 ms). (c) Window size 9×9 (processing time = 266 ms). (d) Window size 11×11 (processing time = 313 ms).

Fig. 12. Comparison of results of different interpolation methods for image expansion. (a) Raw enlarged image. (b) Result of proposed method. (c) Result of simple pixel duplication. (d) Result of 4NN interpolation method. Note that the original image size is 256×256; which is enlarged to 512×512 in this experiment. The cropped versions of the enlarged image are shown here.

(3) region-wise weighting. This method has reasonable processing speed and produces good image quality, compared with other conventional methods, as seen from the experimental results. Future studies may be directed to interpolation of unwarped images of omni-cameras other than the hypercatadioptric type, like the fish-eye camera.

Acknowledgement

This work was supported financially by the Ministry of Economic Affairs under Project No. MOEA 93-EC-17-A-02-S1-032 in the Technology Development Program for Academia.

References

Atwood, G.H., Davis, W.A., 1989. Image expansion using interpolation and heuristic edge following, in: Proc. of 3rd Int. Conf. on Image Processing and its Applications, London, UK, pp. 664–668.

Baker, S., Nayar, S.K., 1999. A theory of single-viewpoint catadioptric image formation. Int. J. Comput. Vis. 35 (2), 175–196.

Jain, A.K., 1989. Fundamentals of Digital Image Processing. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, USA.

Jeng, S.W., Tsai, W.H., 2004. Construction of Perspective and Panoramic Images from Omni-Images Taken from Hypercatadioptric Cameras for Visual Surveillance, in: Proc. of 2004 IEEE Int. Conf. on Networking, Sensing, and Control, Taipei, Taiwan, March 21–23, pp. 204–209.

Li, X., Orchard, M.T., 2001. New edge-directed interpolation. IEEE Trans. Image Process. 10 (10), 1521–1527.

Mashita, T., Iwai, Y., Yachida, M., 2006. Calibration method for misaligned catadioptric camera. IEICE Trans. Inf. Syst. E89-D (7), 1984–1993.

Nayar, S.K., 1997a. "Omnidirectional Vision," in: Proc. of Eighth Int. Symp. on Robotics Research (ISRR), Shonan, Japan.

Nayar, S.K., 1997b. "Catadioptric Omnidirectional Camera," in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, pp. 482–488.

Onoe, Y., Yokoya, N., Yamazawa, K., Takemura, H., 1998. Visual Surveillance and Monitoring System Using An Omni-Directional Video Camera, in: Proc. of 14th Int. Conf. on Pattern Recognit., vol. 1, Brisbane, Australia, pp. 588–592.

Peri, V.N., Nayar, S.K.. 1997. Generation of Perspective and Panoramic Video from Omnidirectional Video, in: Proc. of DARPA Image Understanding Workshop, New Orleans, LA, USA, pp. 243–246.

Ramponi, G., Carrato, S., 2001. An adaptive irregular sampling algorithm and its application to image coding. Image Vis. Comput. 19, 451–460.

Swaminathan, R., Grossberg, M.D., Nayar, S.K., 2001. Caustics of Catadioptric Cameras, in: Proc. of 8th IEEE Int. Conf. on Comput Vision, vol. 2., Vancouver, BC, Canada, July 7–14, pp. 2–9.

Yamazawa, K., Yasushi, Y., Yachida, M., 1993. Omnidirectional Imaging with Hyperboloidal Projection, in: Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Yokohama, Japan, July 26–30, pp. 1029–1034.

Yamazawa, K., Yagi, Y., Yachida, M., 1993. New real-time omnidirectional image sensor with hyperboloidal mirror, in: Proc. of 8th Scandinavian Conf. on Image Analysis, vol. 2., Tromso, Norway, pp. 1381–1387.

Ying, X.H., Hu, Z.Y., 2004. Catadioptric camera calibration using geometric invariants. IEEE Trans. Pattern Anal. Mach. Intell. 26 (10), 1260–1271.