A NOVEL RECTIFICATION METHOD FOR TWO-DIMENSIONAL CAMERA ARRAY BY PARALLELIZING LOCUS OF FEATURE POINTS

Norishige FUKUSHIMA, Tomohiro YENDO, Toshiaki FUJII, and Masayuki TANIMOTO

Graduate School of Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8603, JAPAN
E-mail:fukushima@tanimoto.nuee.nagoya-u.ac.jp, {yendo,fujii,tanimoto}@nuee.nagoya-u.ac.jp

ABSTRACT

In this paper, we present a novel rectification method for two-dimensional multi-camera array. Rectification and correction are important for free view-point imaging and multi-view coding in the field of image-based rendering (IBR). Our rectification process consists of two step; intrinsic and directional parameters trimming and orthogonal warping of image plane. Experimental result show that the pixel error of our rectification become less than 0.25 pixel. In consequence of this, that precision of rectification has enough accuracy for such image based rendering techniques.

Keywords: Rectification, Camera Array, Vanishing Point

1. INTRODUCTION

This paper presents a novel correction method for geometric distortion on two-dimensional multi camera array (2DCA) images which is shown in Fig. 1. The 2-D camera array is a set of huge amount of cameras which are thickly placed and strictly aligned.

First of all, let's clarify the goal of this paper. When we snap the scene by dense 2DCA, we can see lots of common regions in the images. If the 2DCA is ideal condition, lines which connect a points on image to the point on neighborhood multi camera images become square lattice (see in Fig.2 right side). We call this points and lines "*locus of feature points*". In this condition, we can find correspondence easily among images, and it is effective for Multi-View Coding (MVC)[1], Ray-Space[2], and Light Filed Rendering (LFR)[3]. For instance, coding performance could be better on MVC and rendering arbitrary viewpoint image[4] would become faster. Unfortunately, to setup aligned array with high accuracy is impossible. It is because that cameras have individual characteristic and difference, that is hard to correct and inevitable. In addition, setting the camera by human hand must have irregular matters, so that cameras may be settled wrong direction. For this reason, the corresponding points are messed like Fig.2 left side. To conclude above story, our research aim is that we rectify the random motion of points into the fixed grid by image correction. In this paper, we call this geometric error compensation, *Rectification*.

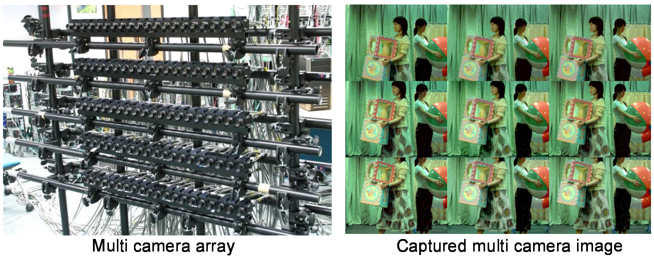


Fig. 1. Camera array

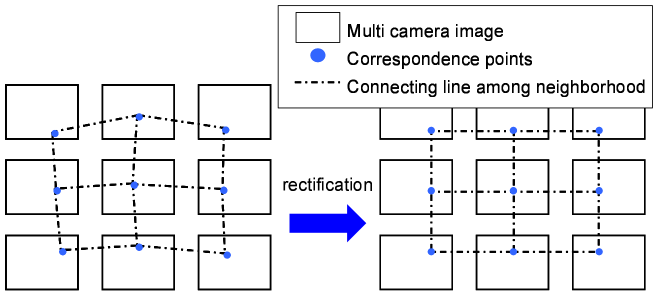


Fig. 2. Locus of Feature Point

In the context of image rectification, these processes are well researched in Computer Vision (CV) as a part of stereo matching problem since early times[5]. To rectify a stereo camera to have parallel epipolar line[6] and a trinocular camera to have orthogonal epipolar one[7] are already studied in CV field. Increasing the interest in IBR, LFR and Multi-baseline stereo, the needs of camera array which consists of more than four cameras images are also increasing.

The rectification method described in this paper belongs to four (or more) camera's 2DCA setup. Whereas the stereo rectification methods have been maturate, methods of rectification for 2DCA did not have large attention, but some works are reported. Vanish have presented a "Plane + Parallax rectification" for 2DCA[8]. This method approximate camera model as affine, hence successful condition is limited. Matsumoto have shown a "Calibrated based camera rectification" for various camera setup[9]. Unfortunately, this method relies hardly on camera calibration[10] accuracy while exact camera calibration is difficult. Deng have proposed a "Light field rectification". This method is suitable for our camera setup except for one critical assumption. It is that they use "one" machine controlled camera platform for setting precise camera position, same intrinsic parameter, and orientation parameter. This condition limits that the system can capture only still image. Therefore enough studies have not tried to our rectification case, such as using multi-camera with 2-D camera setup.

Our method is divided into two steps; all cameras' intrinsic and directional parameter conforming step and image plane othogonalization step. At first step, we align the camera intrinsic parameter and directional parameter by using infinite correspondence points, and then all cameras become same condition except for camera position parameter. Next step, we change the direction of optical axis on camera array to have the condition that the orientation becomes orthogonal for camera array plane. As a result of rectification, the distance from ideal case become 0.25 pixel, and consequence we can make out that the result will come out sufficient for LFR and Ray-Space construction.

This paper is organized as follows; In Section 2 we address the camera condition and setup. We remark our rectification method with detail in Section 3. The experimental result, which includes computer graphics simulation and practical real camera arrays case, are shown in Section 4. Finally, in Section 5 we conclude by noting that this method is enough to construct ray-space.

2. CAMERA ARRAY RECTIFICATION

2.1. Pin-hole Camera Model

Throughout this paper, we assume camera as basic pin-hole camera model[12], and we assume no lens distortion via removing this distortion[13].

Following pin-hole camera model, the 3-D point M is projected to image plane point m by projection matrix which has 11 degree of freedom (DOF);

$$m = K[R;T]M$$

where K is intrinsic parameter , T is position parameter and R is orientation parameter. K is upper triangular matrix and has 5 DOF. T is 3D vector and has 3 DOF. R is orthogonal 3×3 matrix and has 3 DOF. $[R;T]$ is 4×3 projection matrix to map 3D to 2D.

2.2. Ideal 2-D camera array in 3-D domain

Binocular rectification makes the orientation of two cameras same, make it orthogonal to camera baseline, and also two cameras have same intrinsic parameter. On the contrary, in the camera array rectification, all n cameras is satisfied about the above assumptions, where n is the number of camera. In addition, camera orientation are orthogonal to camera array plane P (Fig. 3). Thus, all intrinsic parameters $K_i(i = 1, ..., n)$ are same, and vector of optical axis derived from orientation

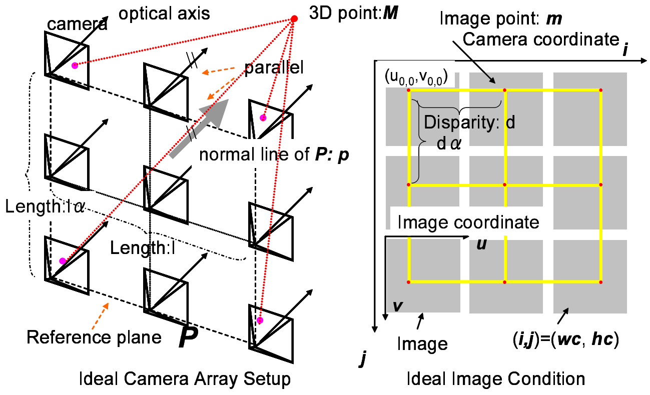


Fig. 3. Ideal condition of rectified camera array

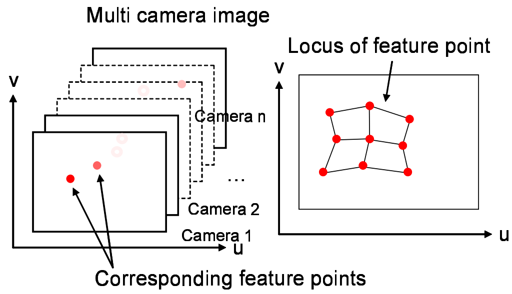


Fig. 4. Another Image Represantation

$R_i(i = 1, ..., n)$ and the normal vector p of camera array plane are also same.

2.3. Ideal 2-D camera array in image domain

In the case of capturing multi-camera images with ideal 2D camera array, which is $w_c col \times h_c row$ matrix setup, the correspondence points have following properties. When a 3D point M project a point m whose image coordinate is $(u_{i,j}, v_{i,j})$ on the camera which place at (i, j) camera coordinate, correspondence points sequences from the same camera array's "row" exist in same image scan-line u. Also points sequences consisted from the cameras posited same camera array "column" exist in same image vertical line v (right side of Fig. 3). Accordingly, the relationship among i, j and u, v is

$$v_{i,1} = v_{i,2} = ... = v_{i,h_c}$$

$$u_{1,j} = u_{2,j} = ... = u_{w_c,j}.$$

In addition, if this camera array is placed at completely square lattice grid, the disparities d of neighborhood correspondence points become same.

$$d = v_{1,j} - v_{2,j} = v_{2,j} - v_{3,j} = ... = v_{cw-1,j} - v_{cw,j}$$

$$d\alpha = u_{i,1} - u_{i,2} = u_{i,1} - u_{i,2} = u_{i,ch-1} - u_{i,ch}$$

$$(u_{i,j}, v_{i,j}) = (u_1 + id, v_1 + j + d\alpha)$$

Where α is the aspect ratio of the camera array position.

2.4. Muti Camera Image Representation

In this paper, we represent multi camera images by two type. One is multi image domain which is seen in Fig. 3. Another one is xxx domain which is shown in Fig. 4. We choose the type of representation which is more suitable for explanations.

3. PROPOSAL RECTIFICATION METHOD

The proposal method consists of two parts; at first, we trim all camera's intrinsic parameter and orientation parameter as same to cancel the individual difference, and next step, we rotate the orientation of principal point vector ,which now all camera orientation matrix are same, to make orthogonal to camera array plane.

3.1. Intrinsic and orientation parameter homogenization

A homography matrix H_{ij} represents a projection of point M on a 3D plane Π into the two image planes m_i, m_j at camera $i, j (i \neq j)$

$$m_i = H_{ij} m_j$$

H_{ij} can be decomposed into, which the book [12] shows,

$$H_{ij} = K_i (R_{ij} + \frac{tn^T}{l}) K_j^{-1}.$$

where R_{ij} is the relative orientation of i and j (i is base orientation), thus $R_{ij} = R_i R_j^{-1}$. t is the translation vector, K_i, K_j are intrinsic parameter of each camera, l is the distance between plane Π and camera baseline, and n is the normal vector of plane Π(Fig. 5).

If the plane posits at infinity distance, the distance l becomes $l \to \infty$, thus the above relationship become simple and is represented by the homography H_{ij}^{∞} on the plane at infinity Π_∞

$$H_{ij}^{\infty} = K_i R_{ij} K_j^{-1}.$$

In this case, we can ignore the translation effect, therefore, we can focus the relation of only intrinsic and orientation parameter. Now we change this matrix representation as follow;

$$K_i R_i = H_{ij}^{\infty} K_j R_j.$$

This equation shows that if we multiply the H_{ij}^{∞} to camera image of j, the j camera and i camera's parameter of orientation and intrinsic become same. In a similar way, we apply multiply this homography to $j = \{1, 2, ..., n\} \backslash i$. "\" means exception of set. After that process, direction and intrinsic characteristic on all cameras become same.

Note that this $H_{i,j}^{\infty}$ require four correspondence point at infinity. It is because that a plane needs 4 point to make square.

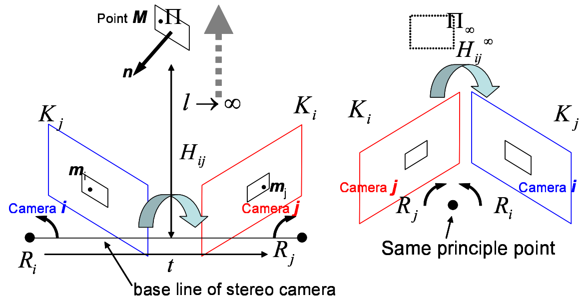


Fig. 5. Relation among image , plane and H

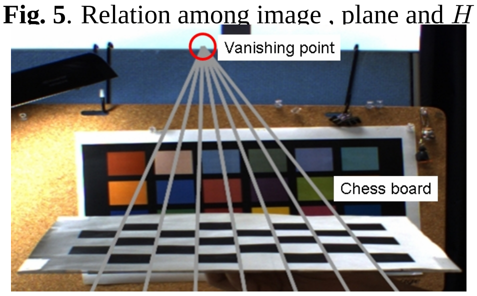


Fig. 7. Vanishing point from chess board

3.2. Vanishing Point and Infinity Point

Plane at infinity are well known as panorama mosaic method for conjugating faraway landscapes scene[15] in recent decade. This method use infinite correspondence points and overlap images by perspective transform (see in Fig. 6). We use that idea to rectify images as infinite points matching. Unfortunately, we cannot always use such far scene for rectification at anytime, thus we also use vanishing points from a plane pattern.

The vanishing point is cross point of two parallel lines in 3-D scene on the perspective camera domain (Fig. 7). This means that the cross-point on these lines is placed at infinity.

In practice, we cannot avoid noise effects at feature points detection so that we cannot define one point as the cross point of plane pattern. Following paragraphs, we are going to address an optimization method for this problem.

Fig. 8 shows the optimization method. At first, we extract an observation feature points p_f on chess board, and then we fit these points to ideal square grid points p_g by ideal grid projection homography matrix H_g. In this process, we compute H_g matrix with Levenberg-Marquardt optimization method[12] which is based on a Newton projection error min-

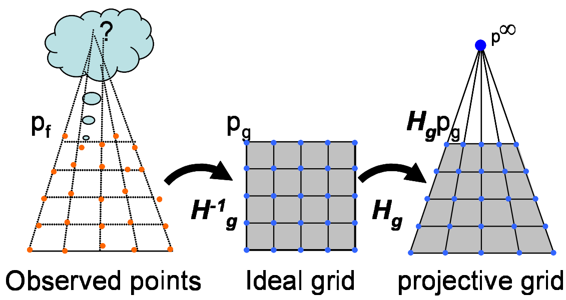


Fig. 8. Vanishing point optimization

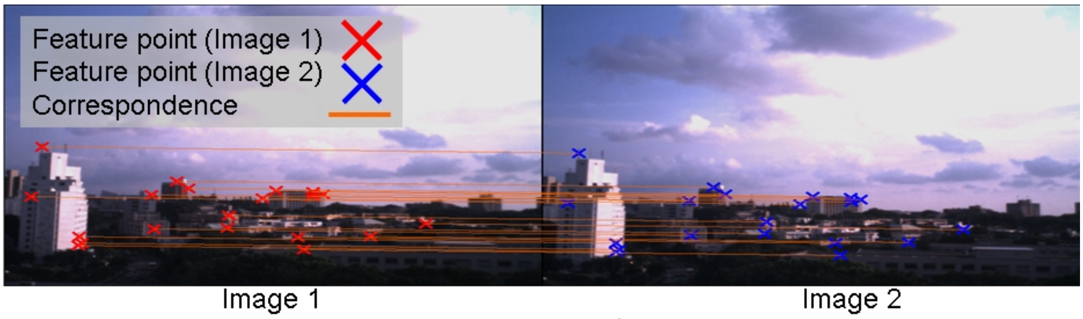


Fig. 6. Corresponding points at Infinity on Far Scenery

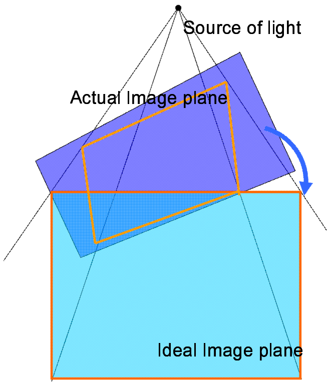


Fig. 9. The Condition of Before Othogonalization

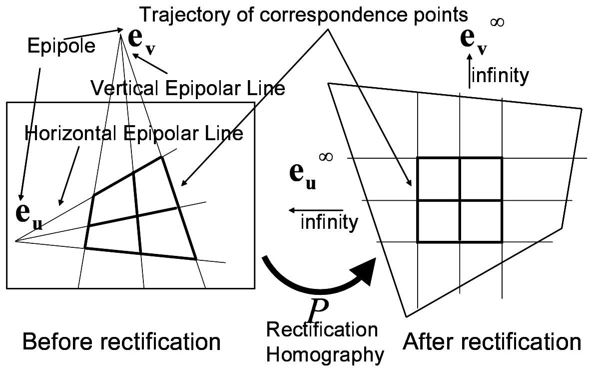


Fig. 10. The Image of Correspondence Point Orbit in the case of array aligned by H^{∞}

imization method. That representation is written by

$$p_f = H_g p_g.$$

Above points of $H_g p_g$ satisfy ideal trapezoid so that the cross point must be one, and then this point is used as infinite point p_i^{∞} on i camera.

Next step, we compute the infinite homography H^{∞} by using these infinite correspondence points. the homography H_{ij}^{∞} adjusting camera j to base camera i are solved by more than four infinite correspondence points.

$$p_i^{\infty} = H_{ij}^{\infty} p_j^{\infty}$$

After that, we compute this homography at each n camera to project i camera one by one.

In this step, there is one notification that the correspondence infinite points have outliers in case, using natural infinite points and vanishing points on chess board. In former case, we cannot avoid miss matching in auto correspondence detection. Also in latter case, we cannot avoid the large error of detection for vanishing point because these infinite points estimation process is indirect method. Thus, we solve this homography matrix by LMedS (*L*east *Med*ian *S*quare) method [14] because of the robustness of outlier.

3.3. Othogonalization of Camera Array

After the semi-rectification of previous subsection, the camera array has already had some regularity, except for the direction of image plane. Fig. 9 explains this condition. There is a square pyramid which is composed by a point of light source as an apex and a camera array plane as a bottom square, and then the volume is cut off by the image plane of camera array. If the image plane is parallel to camera array plane, the cross-section of light pyramid becomes quadrate. Actually, the cut-plane indicates the locus of feature points. On the contrary, the placed camera array has most always wrong direction, hence the direction of array depends on the base camera direction described in previous subsection. Consequently cutting surface has perspective distortion. Therefore we change the direction of optical axis to cut with straightforwardness. Fig. 10 shows a locus of a corresponding point with overlapping domain with two case; deal and distorted.

By the way, quadrangle, which consists of four non-parallel lines, has two intersecting points in general. We call this intersection "epipole of camera array". This intersection is found by the extracting method of vanishing point over chessboard similarly. For making tetragonal lattice, the two epipoles (vertical and horizontal one) must be projected to infinity respectively. It is because that intersection of parallel lines place at infinity to make square. More noteworthy is that there are as many locuses as number of corresponding feature points, so that there are a lot of epipoles due to disturbing noise. Thus we take on a center of mass as a representative epipole.

We write the horizontal epipole $e_u = (e_{u0}, e_{u1}, 1)^T$ and infinite one $e'_u = (1, 0, 0)^T$ by homogeneous vector. In a similar way, horizontal things are $e_v = (e_{v0}, e_{v1}, 1)^T$ and $e'_v = (0, 1, 0)^T$. Therefore the matrix P which projects the epipoles to infinity is as follow;

$$e'_i = Pe_i, i \in u, v$$

In this representation, it has shifting and expanding ambiguity, so detail projection matrix P^* becomes

$$[e'_u e'_v] = P^*[e_u e_v]$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & a & 0 \\ b & 1 & 0 \\ p_1 & p_2 & 1 \end{bmatrix} \begin{bmatrix} e_{u0} & e_{v0} \\ e_{u1} & e_{v1} \\ 1 & 1 \end{bmatrix}.$$

The parameters of P are solved direct linear solution. After solving of P^*, all corresponding point become parallel at each direction, horizontal and vertical. It ,however, has still degree of freedom, such as mirroring, aspect ratio, zooming, and shifting. Thus we define the mirroring cancel matrix A and correcting other factor matrix S to define this latitude, and are as follow.

$$A = \begin{bmatrix} m & 0 & 0 \\ 0 & n & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S = \begin{bmatrix} s & 0 & \mathrm{shift_u} \\ 0 & s\alpha & \mathrm{shift_v} \\ 0 & 0 & 1 \end{bmatrix}$$

where $m, n \in \{1, -1\}$ is the flip control parameter, s is scaling parameter, and $\mathrm{shift_u}, \mathrm{shift_v}$ are shifting parameters at each u,v image coordinate. The parameter of α is the aspect ratio of camera array, which is known on ahead. Moreover, the aspect ratio of correspondence locus becomes same for this array shape, so that this aspect ratio parameter is settled. parameter s and $\mathrm{shift_u}, \mathrm{shift_v}$ are determined for maximize a valid area of base camera. the flipping parameter $m, n \in \{1, -1\}$ are defined to preserve the original image.

Finally, we can obtain the matrix

$$P = SAP^*$$

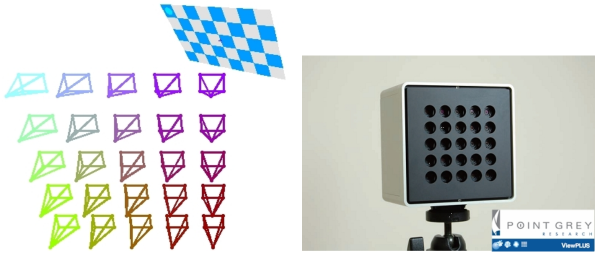
which change the image plane direction to front parallel. As a result, the final rectification homographies H_i for each camera i are

$$H_i = PH_i^{\infty}.$$

4. EXPERIMENTAL RESULT

4.1. Experimental Environment

We have evaluated our method by Computer Graphics (CG) simulation and practical camera array. Fig. 11 shows simulating condition and the real camera array for this experiment respectively. In each experiment, the image resolution is 640×480, and field of view are 36.8°. The camera array are consisted of $5 \times 5 (i_{max} \times j_{max})$ cameras and each pitch is 1 cm. In this experiment, a chess board, which has $35cm \times 20cm$ pattern, has 1m distance from the array. For compute the infinity points, we capture the plane pattern which has 7×4 grid at 20 times.($7 \times 4 \times 20 = n_{max}$ points).



(a) simulated 5×5 camera array (b) real 5×5 camera array

Fig. 11. Experimental environment

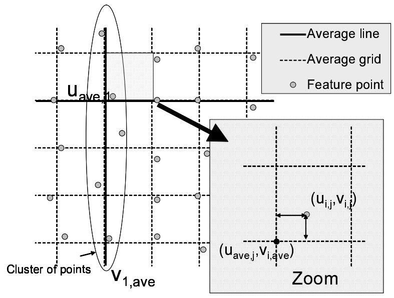


Fig. 12. Error function of rectification

In the CG case, the orientation of each camera are settled with uniform distribution random number ($| \leq 2^{\circ}|$) at pitch, yaw, and roll coordinate. focal length as intrinsic parameter on each camera has random difference up to 50 pixels. The error of extracted chess board points are modelized by Gaussian noise, whose average is 0 and standard variation is σ_u, σ_v along the image coordinate u, v.

4.2. Error Function

The rectification error is defined by pixel distance and written in detail in this section. Fig. 12 shows a sample of rectified feature projection points of a 3D scene. a feature point on the pattern $k(K = 1, 2, ..., n_{max})$ at camera line i and camera line j is written by $u^k_{i,j}, v^k_{i,j}$.

Average image coordinate of line j at u is represented by $\overline{u}^k_{ave,j} = \frac{1}{j_{max}} \sum_j u_{i,j}$ and same line i and v is $\overline{v}^k_{i,ave} = \frac{1}{i_{max}} \sum_i v_{i,j}$. The normalized distance of u, v coordinate is written by E_u, E_v,

$$E_u = \sqrt{\frac{1}{n_{max} \times i_{max} \times j_{max}} \sum_n^{n_{max}} \sum_j^{j_{max}} \sum_i^{i_{max}} (\overline{u}^n_{ave,j} - u^n_{i,j})^2}$$

$$E_v = \sqrt{\frac{1}{n_{max} \times i_{max} \times j_{max}} \sum_n^{n_{max}} \sum_j^{j_{max}} \sum_i^{i_{max}} (\overline{v}^n_{i,ave} - v^n_{i,j})^2}$$

4.3. Simulated and Practical Experimental Result

At first, we show the simulated experimental result. The graph Fig. 13 indicates noise-resistance rate of three case; ideal, using direct infinite point method and using second hand vanishing point one. The vertical axis is the distance [pixel]E from

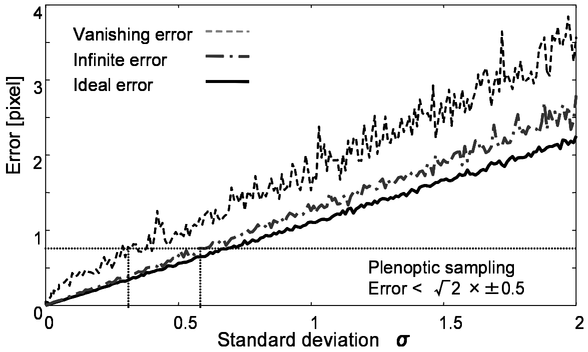


Fig. 13. The relation between errors of pixels and variance of noise by ideal and proposed method

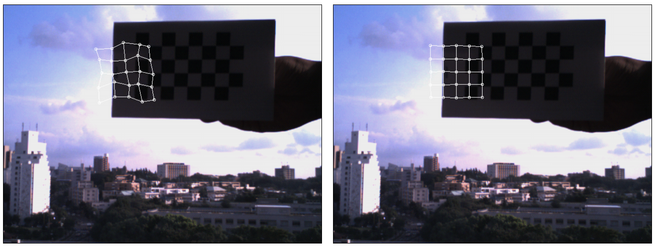


Fig. 14. Images of before rectification and after rectification

ideal grid, which is defined in previous subsection. ($E = \sqrt{E_u^2 + E_v^2}$. The horizontal axis is standard variation of feature point on chess board pattern, written by $\sigma_u = \sigma_v = \sigma$.

We can see that the method using direct infinite point is better than the other from Fig.13. It has still more error than ideal case, however. From the aspect of Plenoptic Sampling[16],which is the theory of image-based sampling, we need to suppress distance within ± 0.5 pixel at each u, v coordinate, so that, E must be $\sqrt{2} \times 0.5$ or less. In this case, chess board method allows 0.3 pixel noise-level and then infinite point method does 0.55 pixel one.

Why using natural point at infinity has better result than using vanishing point over chess board is that, extracting feature point in far scene directly includes less error than indirect method. the chess board method contains large error when the pattern places front parallel to the camera array.

Next step, we present a practical experimentation. Using Fig. 9b camera array, the distance of no-rectification image was 21.52 pixels. The rectification method using vanishing points from 20 chessboards make this error 0.91 pixel. In addition, direct method using infinite point on a far scene in Fig. 14 become 0.25 pixel, when correspondence point of far point are given by SIFT (Scale Invariant Feature Transform) method[17].Fig. 14 is the figure of input picture and rectified picture with locus of feature points.

5. CONCLUSION

In this paper, we have presented the rectification method for 2-D camera array via correspondence points parallelizing. This method requires correspondence of 4 infinite points and 1 finite point among multi camera images at least. Experimental result of real camera array rectification shows that, the geometric distance of after rectification becomes 0.25 pixel, it shows that it has enough accuracy in practical. As a contribution of this paper, we can produce the precisely rectified multi-camera sequences. It helps IBR and MVC researchers, who are eager to obtain fine multi-view materials, carry on there work.

Proposed method can rectify the 2D camera array images only by correspondence of natural feature points if infinite points are depicted. However, the necessity of infinity becomes week point of this method. It is because the process divides our procedure into two steps, as a result error optimization between those two steps does not work well. Thus, the limitation of optimization still exists. Therefore we will research a new method requiring the no infinite points.

6. REFERENCES

[1] K. Yamamoto, T. Yendo, T. Fujii and M. Tanimoto, ”Multi-view Video Coding using Viewinterpolated Reference Images “, Proc.PCS2006, SS3–1 (2006).

[2] T. Fujii, T. Kimoto, M. Tanimoto, “Ray Space Coding for 3D Visual Communication“, Picture Coding Symposium ’96, pp. 447–451, (1996).

[3] M. Levoy, P Hanrahan, “Light Field Rendering”, Proc. ACM SIGGRAPH ’96, pp. 31–42 (1996).

[4] N. Fukushima, T. Yendo, T. Fujii and M. Tanimoto, “Free View-point Image Generation Using MultiPass Dynamic Programming”, Proc.SPIE SD&A XIV,vol.6490,pp. 460–470 (2007).

[5] N. Ayache, C. Hansen, ”Rectification of images for binocular and trinocular stereovision” 9th International Conference Pattern Recognition 1988, vol.1, pp. 11–16 (1988).

[6] C. Loop and Z. Zhang, ”Computing rectifying homographies for stereo vision“ Proc. IEEE Conf. CVPR1999, Vol. 1, pp. 125 – 131 (1999).

[7] L. An, Y. Jia, J. Wang, X. Zhang, and M. Li, ” An efficient rectification method for trinocular stereovision”, Proc. Int. Conf. on Pattern Recognition, vol. 4, pp. 56–59, (2004).

[8] V. Vaish, B. Wilburn, N. Joshi, M. Levoy, ”Using plane + parallax for calibrating dense camera arrays ” Proc. CVPR 2004, val. 1, pp. I-2 – I-9 (2004).

[9] K. Matsumoto, T. Yendo, T. Fujii and M. Tanimoto, ”Geometric Correction of Multi-Camera Images for 3D-Image Processing”, Proc. of The 10th Image Media Processing Symposium, pp. 89–90 (2005).

[10] Z. Zhang, ”A Flexible New Technique for Camera Calibration“ , IEEE Trans. PAMI, Vol.22, No.11, pp.1330–1334 (2000).

[11] K. Deng, L. Wang, Z. Lin, T. Feng, and Z. Deng, ” Correction and rectification of light fields” In Computers & Graphics. , 27(2), pages 169–177 (2003).

[12] R. Hartley and A. Zisserman, ”Multiple View Geometry in Computer Vision Second Edition” Cambridge University Press, Cambridge, UK (2003).

[13] R Swaminathan, SK Nayar, ”Nonmetric calibration of wide-angle lenses and polycameras” IEEE Trans. PAMI, Vol. 22, No. 10, pp. 1172 –1178 (2000).

[14] P.J. Rousseeuw, ”Least Median of Squares Regression” J. American Stat. Assoc., Vol.79, pp. 871–880 (1984).

[15] R. Szeliski, ”Video mosaics for virtual environments” IEEE Computer Graphics and Applications, pp. 22–30, (1996).

[16] JX Chai, X Tong, SC Chan, HY Shum, ”Plenoptic sampling”, Proc. SIGGRAPH ’00, pp.307–318, (2000)

[17] David G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, 60, 2 , pp. 91-110 (2004).