IPSJ Transactions on Computer Vision and Applications Vol. 2 132–144 (Nov. 2010)

Research Paper

Object-Oriented Color Calibration of Multi-viewpoint Cameras in Sparse and Convergent Arrangement

Shohei Nobuhara,[†1] Yuta Kimura[†1,*1]
and Takashi Matsuyama[†1]

This paper proposes a new color calibration method for multi-viewpoint images captured by sparsely and convergently arranged cameras. The main contribution of our method is its practical and efficient procedure while traditional methods are known to be labor-intensive. Because our method automatically extracts 3D points in the scene for color calibration, we do not need to capture color calibration objects like Macbeth chart. This enables us to calibrate a set of multi-viewpoint images whose capture environment is not available. Experiments with real images show that our method can minimize the difference of pixel values (1) quantitatively by leave-one-out evaluation, and (2) qualitatively by rendering a 3D video.

1. Introduction

This paper is aimed at presenting a new color calibration algorithm for multi-viewpoint images captured by sparsely and convergently arranged cameras (**Fig. 1**). This type of camera arrangement is used with the intrinsic and extrinsic camera parameters for 3D shape reconstruction of moving objects [1),2)] or 3D shape reconstruction from Internet photos [3)], for example.

Our method (1) estimates 3D points in the scene robustly and accurately from multi-viewpoint wide-baseline images, and then (2) estimates lens vignetting and imager response parameters of all cameras simultaneously in order to match the colors of the estimated 3D points projections.

The contribution of this algorithm is its convenience and practicality. Our method does not require known calibration objects because we estimate parameters directly from the original images. Therefore we do not need to carry out additional capture of calibration objects (Macbeth color checker, for example) which should be done very carefully and hence is known to be time-consuming and labor-intensive work. In addition, our method can be used for multi-viewpoint images whose capture environment is not known or lost. This enables us to estimate the calibration parameters of image sets collected from the Internet [3)] or images taken before.

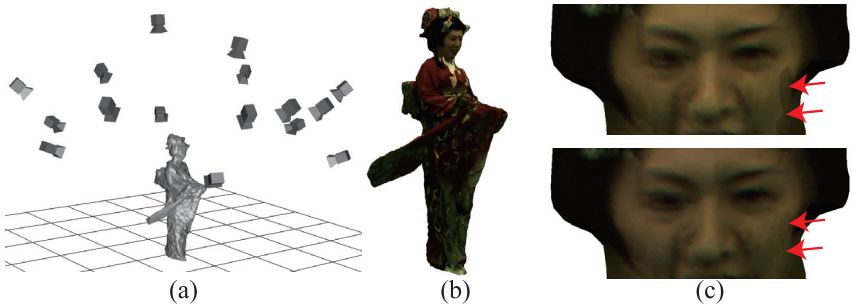


Fig. 1 Color calibration of sparsely and convergently arranged cameras. (a) Cameras capturing an object, (b) Texture mapping result with an estimated 3D shape of the object, (c) Close-up images. Top: without color calibration. Bottom: with our color calibration. The red arrows indicate the boundaries of mapped texture where a texture-source camera switches to another, and therefore we can check the color consistency there.

However our method is not designed to estimate the "true" colors of the scene because we do not have any knowledge about the real colors of 3D points and just make the colors "consistent" to each other. This is a clear limitation of our method, and therefore our method does not advance the state-of-the-art of color calibration in quality.

The rest of this paper is organized as follows. Section 2 briefly reviews related work. Section 3 describes the first step of the algorithm to obtain 3D points in the scene robustly and accurately. Section 4 defines the model of the lens and the camera imager. Section 5 proposes our color calibration algorithm and Section 6 shows qualitative and quantitative evaluations.

2. Related Work

Many studies on multi-viewpoint color calibration have been proposed so far.

†1 Kyoto University
*1 Presently with Canon Corporation

The literature can be categorized into two groups. The first group consists in calibrating imager responses only [4)–7)], and the second group consists in calibrating imager responses and the vignetting effect simultaneously [8)–10)].

Our proposed algorithm can be categorized in the second group, nevertheless it has two differences. The first point is the camera arrangement. Conventional algorithms which calibrate lens vignetting effects and imager responses simultaneously assume that the optic axes of the multiple cameras are almost parallel so that they can expect that two contiguous images share a certain amount of overlaps [8)–10)] at the pixel level. On the other hand, we only assume that two contiguous images share some image features. This enables us to apply our algorithm for cameras in a sparse and convergent arrangement (Fig. 1 (a)). The second point is calibration objects. Unlike other approaches, our method uses image features in the original images, and does not require known objects like "Macbeth color checker". Therefore we have an advantage on its applicability. In particular, we can even apply our method to datasets taken before and cannot capture such images. However, this point implies that our calibration cannot produce "true" colors, and just makes colors consistent to each other.

Besides there is another approach that calibrates the lens vignetting effect from a single image [11)]. This approach does not depend on the camera arrangement and therefore can be combined with methods without vignetting correction. However this approach assumes that the radiometric shading is the dominant shading in input images. Hence this assumption cannot handle images taken by multi-viewpoint cameras in a studio lit by near light sources.

3. Robust and Accurate 3D Point Estimation from Wide-baseline Camera Images

The algorithm we propose here aims at providing corresponding points between multi-viewpoint images. These corresponding points will be used to define an objective function for our color calibration in Section 5. While several feature points extraction and matching algorithms for wide-baseline non-parallel cameras have been proposed so far [12)–14)], we propose the following algorithm which introduces a bi-directional uniqueness constraint to the algorithm by Furukawa [15)] in order to improve the accuracy and robustness of the matching. This is because we can utilize the intrinsic and extrinsic camera parameters for matching explicitly.

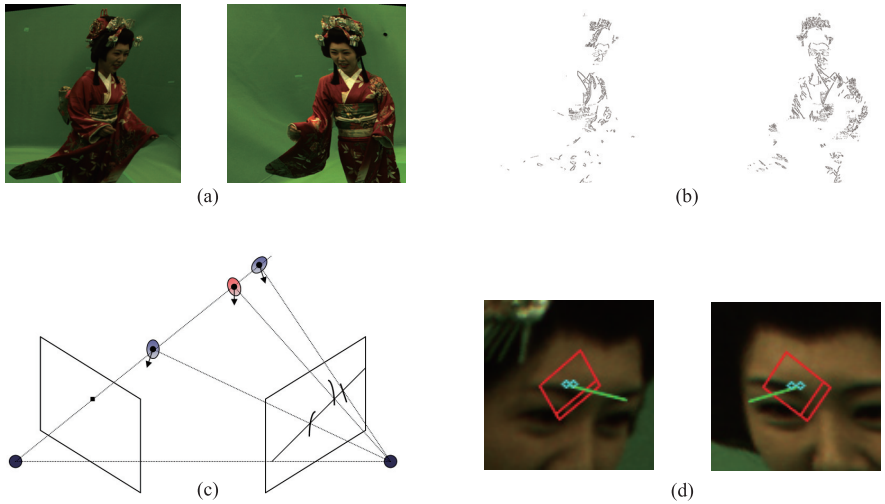


Fig. 2 Matching based on edge features. (a) rectified images, (b) edge features crossing epipolar lines, (c) texture similarity computation with normal direction optimization, (d) an example of matched pair. In (d), the red rectangles illustrate the window used to compute the texture similarity, the green lines illustrate the normals, and the blue circles illustrate the points on either side of the edge feature.

(1) Extract edge features from the images taken by two cameras c and c'.

(2) Eliminate edge features which do not cross the epipolar lines. Let I_E and I'_E denote the resultant images (**Fig. 2** (b)). Here the epipolar line at c' defined by a point e of the edge features in I_E can cross several points of edge features in I'_E. Let $E' = \{e'_j | j = 1, \ldots, n\}$ be the set of these points.

(3) Compute the texture similarities between e and $e'_j \in E'$ using the normal direction optimization [15)] (Fig. 2 (c)). Let $\hat{e}'_j$ denote the point in E' which gives the best similarity. To enforce the uniqueness constraint, we accept the pair e and $\hat{e}'_j$ if the similarity between them is significantly better than the one of the second best pair. Otherwise we reject this pair and leave e without correspondence in the camera c' image because matchings are ambiguous. Here the computation of the similarity should be robust to the lens vignetting and imager responses. We used the "Zero-mean

Normalized Cross Correlation" function ZNCC(·) which is known to satisfy this requirement:

$$\begin{aligned}\text{Similarity}(e, e'_j) &= \text{ZNCC}(e, e'_j) \\ &= \frac{\sum_{i=1}^{N}\left\{(I_i - \bar{I}_e)\cdot(I'_i - \bar{I'}_{e'_j})\right\}}{\sqrt{\sum_{i=1}^{N}(I_i - \bar{I}_e)^2}\sqrt{\sum_{i=1}^{N}(I'_i - \bar{I'}_{e'_j})^2}},\end{aligned} \tag{1}$$

where I_i and I'_i are the i-th pixel values in the matching window of e and e'_j of the camera c and c', and $\bar{I}_e$ and $\bar{I'}_{e'_j}$ are the averages of pixel values given by $\bar{I}_e = \frac{\sum_{i=1}^{N} I_i}{N}$ and $\bar{I'}_{e'_j} = \frac{\sum_{i=1}^{N} I'_i}{N}$.

(4) Validate the uniqueness of the correspondence in the opposite direction ($\hat{e}'_j \to e \in E$). If there is another edge feature point in I_E that has a comparable similarity with $\hat{e}'_j$, reject this pair.

(5) The points of the accepted pair lie on the edge features in I_E and I'_E by definition, and therefore the pixel values on them are ambiguous because we cannot tell which points fall on either side of the edge feature. Hence we use two points which exist on either side of the edge feature, not the point on the edge feature (Fig. 2 (d)).

(6) By iterating the steps from (1) to (4) for all $e \in I_E$, we obtain the set of corresponding points between camera c and c'. We denote this set $P_{c,c'} = \left\{\langle p^i_c, p^i_{c'}\rangle \mid i = 1, \ldots, n_{c,c'}\right\}$, where $\langle p_c, p_{c'}\rangle$ is a corresponding pair and $n_{c,c'}$ denotes the number of obtained correspondences.

This algorithm produces a set of corresponding points between two cameras and is applied to all the possible camera pairs. Note that $P_{c,c'}$ is equal to $P_{c',c}$ since we enforce the bi-directional uniqueness constraint.

4. Lens and Imager Response Model

In this paper we model the ideal intensity x of a pixel at a distance r from the optic center of the imager as mapped to the observed intensity x' by the following function:

$$x' = G^i(V(x, r)), \quad i = \{R, G, B\}, \tag{2}$$

where G^i denotes an imager response of R, G, B channel and $V(x, r)$ is the lens vignetting function described below. This function models the ideal intensity as distorted first by the lens according to the distance from the optic center and then distorted by the imager regardless of the radial distance but according to the color channel.

4.1 Lens Vignetting Model

While there are many vignetting models in the literature [16)–18)], we use a simplified Kang-and-Weiss model [11)] for simplicity:

$$V(x, r) = V(r)x, \tag{3}$$

$$V(x, r) = V_G(r)V_A(r)x, \tag{4}$$

$$V_G(r) = (1 - ar), \tag{5}$$

$$V_A(r) = \frac{1}{(1 + (r/f)^2)^2}, \tag{6}$$

where $V_G(r)$ models a geometric factor by a first order polynomial with a coefficient a, and $V_A(r)$ represents an off-axis illumination factor using the focal length of the camera denoted by f.

In this paper we assume that the intrinsic parameters including the focal length are known, but we use another f for this model to keep the applicability of this lens model as much as possible. Therefore our model has two parameters a and f for each camera.

4.2 Imager Response Model

We model the imager response simply by a linear function for each color channel:

$$G^i(x) = \alpha^i x + \beta^i, i = \{R, G, B\} \tag{7}$$

As the result of this modeling we have six parameters to estimate for each camera. Here we do not consider the non-linear gamma correction done in the imager as we assume machine vision cameras which can explicitly disable this function. Note that we may omit the superfix R, G, or B hereafter when not necessary.

5. Color Calibration Algorithm

Using the model from the previous section, we formulate the multi-viewpoint color calibration problem as a non-linear optimization problem which estimates the best parameters of the model which minimize the sum of color differences between corresponding points given by the algorithm in Section 3.

5.1 Color Correction Function

From Eq. (2), we have the inverse function which maps a real intensity x' to the ideal intensity x:

$$x = F_c(x') = \frac{(x' - \beta_c^i)(1 + (r/f_c)^2)^2}{\alpha_c^i(1 - a_c r)}, \quad i = \{R, G, B\}, \tag{8}$$

where f_c, a_c, α_c and β_c denote f, a, α and β of camera c respectively.

5.2 Objective Function

The function to compute the color differences after color correction of a pair of corresponding points $\langle p_c^i, p_{c'}^i \rangle$ given by two cameras c and c' is

$$\begin{aligned} E(\langle p_c^i, p_{c'}^i \rangle) &= \rho\left(F_c(x_{p_c^i}) - F_{c'}(x_{p_{c'}^i})\right) \\ &= \rho\left(\frac{(x_{p_c^i} - \beta_c)(1 + (r/f_c)^2)^2}{\alpha_c(1 - a_c r)}\right. \\ &\quad \left. - \frac{(x_{p_{c'}^i} - \beta_{c'})(1 + (r/f_{c'})^2)^2}{\alpha_{c'}(1 - a_{c'} r)}\right), \end{aligned} \tag{9}$$

where $x_{p_c^i}$ and $x_{p_{c'}^i}$ denote the observed intensity at p_c^i and $p_{c'}^i$. $\rho(\cdot)$ denotes a robust estimator, such as the pseudo-Huber function [19]:

$$\rho(\delta) = 2b^2\left(\sqrt{1 + \frac{\delta^2}{b^2}} - 1\right) \tag{10}$$

where b is a parameter determined by heuristics.

With this function we define an $n_{c,c'}$ dimension error vector of the color differences between cameras c and c' for each color channel as

$$E_{c,c'} = \left(E(\langle p_c^1, p_{c'}^1 \rangle), \ldots, E(\langle p_c^{n_{c,c'}}, p_{c'}^{n_{c,c'}} \rangle)\right)^\top, \tag{11}$$

where $P_{c,c'}$ denotes $n_{c,c'}$ corresponding points between c and c' as defined in Section 3. Finally by concatenating $E_{c,c'}$ over all $\frac{N(N-1)}{2}$ pairs from N cameras, we define the total error function E_0 by

$$E_0 = \begin{pmatrix} E_{1,2}^\top & E_{1,3}^\top & \ldots & E_{N-1,N}^\top \end{pmatrix}^\top. \tag{12}$$

Note that here E_0 does not depend on the order of cameras since the bi-directional constraint in Section 3 makes $P_{i,j}$ equal to $P_{j,i}$ and therefore $E_{i,j}$ is equal to $E_{j,i}$.

This E_0 measures how the parameters can reduce the color differences of a channel across multiple cameras. However $E_0 \to \min$ has a trivial solution $\alpha_c = \infty$ which makes all pixels to be saturated. Therefore we introduce a constraint term which represents the fidelity to the original pixel values. Suppose we have n_c corresponding points for camera c among all the other $N - 1$ cameras and the j-th point of them is denoted by p_c^j. We define the fidelity term of a color channel as

$$D_0 = \begin{pmatrix} D_1^\top \ldots D_c^\top \ldots D_N^\top \end{pmatrix}^\top, \tag{13}$$

where D_c denotes the per-camera fidelity function defined as

$$D_c = \left(D_c(p_c^1), \ldots, D_c(p_c^{n_c})\right)^\top. \tag{14}$$

Here $D_c(p_c^j)$ is a point-wise fidelity function and in this paper we used the following squared error function:

$$D_c(p_c^j) = \left\| x_{p_c^j} - F_c(x_{p_c^j}) \right\|^2. \tag{15}$$

By integrating E_0 and D_0 from all color channels with a weighting factor λ we obtain the final objective function to be minimized as

$$E = \begin{pmatrix} E_0^{R\top} & E_0^{G\top} & E_0^{B\top} & \lambda D_0^{R\top} & \lambda D_0^{G\top} & \lambda D_0^{B\top} \end{pmatrix}^\top. \tag{16}$$

The above-defined objective function E (Eq. (16)) has $8N$ parameters to estimate for N cameras. We assume that we can obtain at least 8 corresponding points for each camera by the algorithm described in Section 3, and estimate these parameters using Levenberg-Marquardt method. Here we use the following initial values $\alpha_c = 1$, $\beta_c = 0$, $a_c = 1$, $f_c = \tilde{f}_c$, where $\tilde{f}_c$ denotes the focal length of the intrinsic parameter of camera c.

6. Evaluation

6.1 Evaluation Using Reference Object

First, in order to evaluate the color calibration process without the corresponding point estimation step, we estimated the color calibration parameters using a reference object from which we can obtain corresponding points without errors

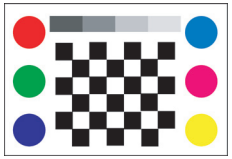


Fig. 3 A color chart as known reference object.

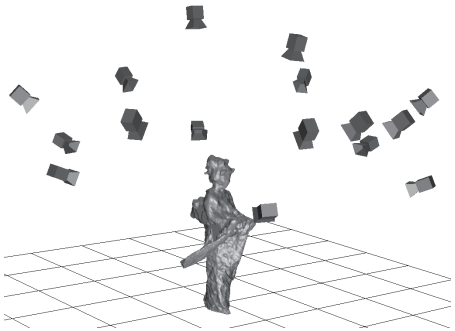


Fig. 4 Camera arrangement.

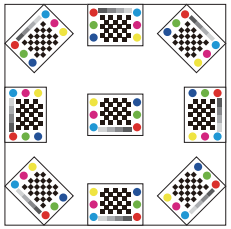


Fig. 5 Arrangement of the reference object (top view).

(**Fig. 3**). The six centers of the colored circles and the four centers of grayscale blocks are extracted as corresponding points. We used 16 UXGA cameras (Point-grey Grasshopper with Kowa LM5JC1M lens) running at 30 fps as shown in **Fig. 4**, and captured the reference object 9 times to uniformly cover the entire region of the scene (**Fig. 5**). **Figure 6** shows a part of the captured images. In

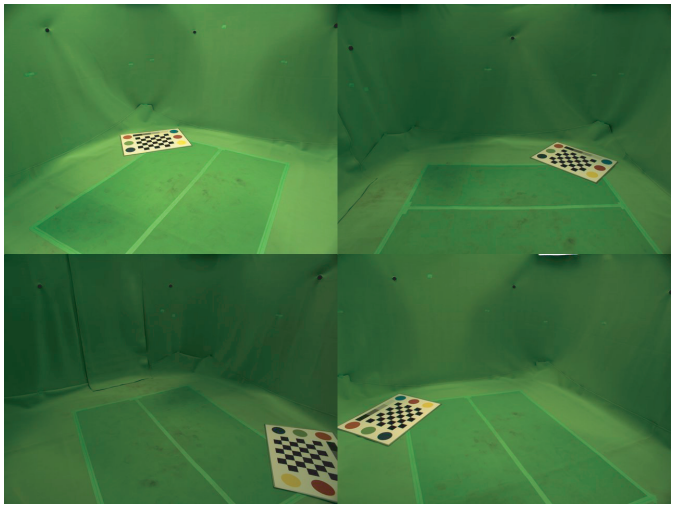


Fig. 6 Examples of captured images.

this experiment, the processing time was approximately 5 minutes by an Intel Core-i7 860 PC.

Figure 7 shows the result of color calibration. The top-left image shows one of the input original images and the top-right image shows the calibrated top-left image. The bottom-left and bottom-right images show input and calibrated images of all 16 cameras. Note that each image is projected to a virtual viewpoint so that we can compare the images easily. From these results, especially focusing on the images with the red borders in Fig. 7, we can observe that we can improve the color consistency across multi-viewpoint cameras qualitatively. **Figure 8** shows the distributions of color differences in [0:255] at corresponding points before and after the color calibration. The difference is computed as $\|{E_0^R}^\top \, {E_0^G}^\top \, {E_0^B}^\top\|$. We can conclude that color calibration with the estimated parameters can reduce the total color differences quantitatively.

6.1.1 Distribution of Corresponding Points

Secondly we conducted an experiment with non-uniformly distributed reference objects as illustrated in **Fig. 9** in order to evaluate how the spatial distribution of the corresponding points affect the calibration result. In the case of Fig. 9 (a)

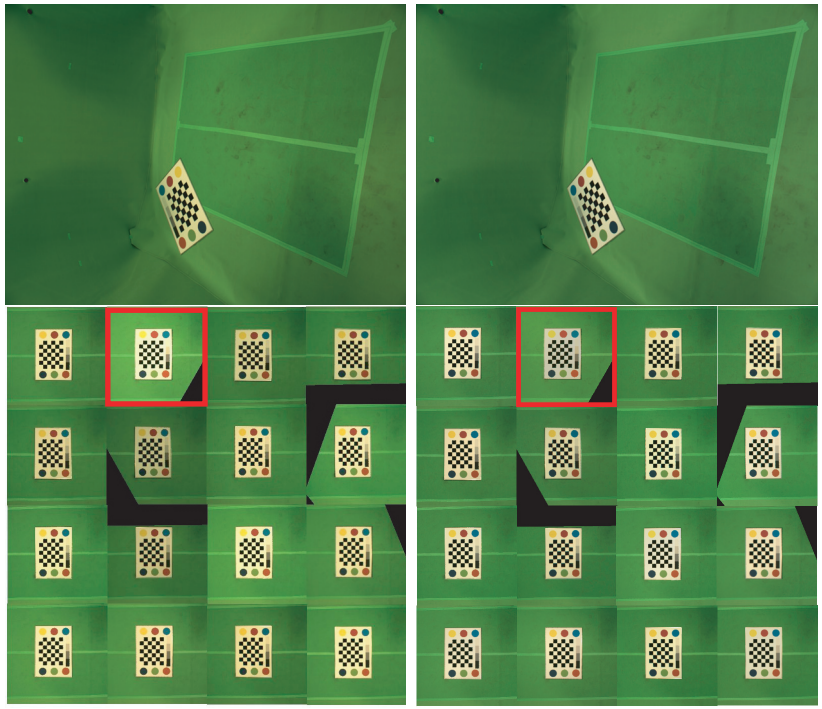


Fig. 7 Color calibration results. Left: without calibration. Right: with calibration. Top: camera images. Bottom: images projected to a virtual viewpoint. Images in the bottom row are projected to a virtual viewpoint so that we can compare them easily. Images with the red borders indicate a camera corrected clearly.

(Pattern A) corresponding points given by the reference object cover the scene as same as the case of Fig. 5 but with a sparse distribution. On the other hand, corresponding points cover a limited area of the scene, i.e., the points distributed non-uniformly in the case of Fig. 9 (b) (Pattern B).

Figures 10 and **11** show the calibrated images of both cases. We can observe that non-uniformly distributed points (Pattern B) give a clearly inappropriate result as marked with red borders in Fig. 11 (b). **Table 1** shows the average differences of colors at all corresponding points in the nine reference objects. These values also prove that we cannot improve the color consistency with non-

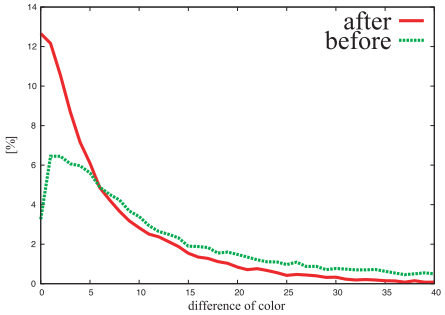
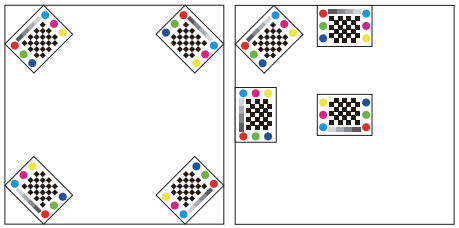


Fig. 8 Color differences in [0:255] on corresponding points.



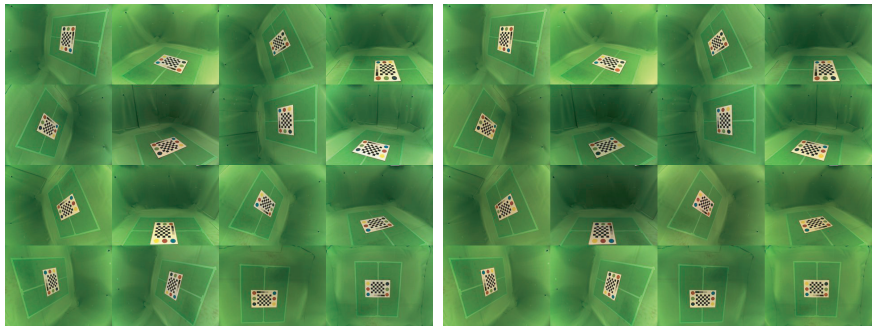
(a) Pattern A (b) Pattern B

Fig. 9 Non-uniform distribution of calibration points.

uniformly distributed points (Pattern B) quantitatively. From these observations we can conclude that the uniform distribution of corresponding points is more important compared with the number of points.

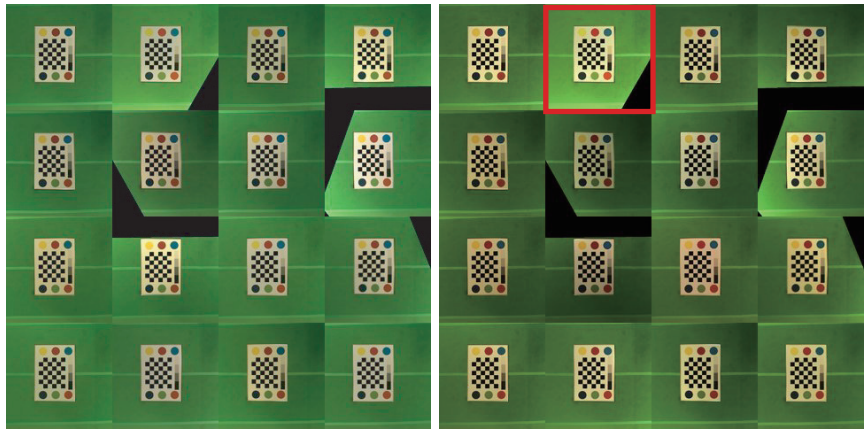
6.1.2 Leave-one-out Evaluation with Parameter λ

Figure 12 illustrates the relationship between (1) numbers of overflowed or underflowed pixels (red) and (2) calibration errors with respect to the parameter λ in Eq. (16) (green). Here we estimated the calibration parameters for each λ using eight reference objects (Pattern C in **Fig. 13**), and used the last reference object to compute the average color difference (Pattern D in Fig. 13). The blue line in the graph shows the error without color calibration. In this graph we can observe that a smaller λ ($\lambda \approx 0$) can minimize the error but increase the number of overflowed or underflowed pixels. This is because the object function



(a) Pattern A (Fig. 9 (a)) (b) Pattern B (Fig. 9 (b))

Fig. 10 Calibration results with non-uniformly distributed points.



(a) Pattern A (Fig. 9 (a)) (b) Pattern B (Fig. 9 (b))

Fig. 11 Calibration results with non-uniformly distributed points. Images are projected to a virtual viewpoint for comparison.

(Eq. (16)) can make all the pixels saturated as mentioned earlier (**Fig. 14** (a)). On the other hand, by comparing Fig. 14 (b), (c) and (d), we can observe that a larger λ ($\lambda > 1$) prevents the calibration process from minimizing the color differences. From these observations we can conclude that (1) choosing λ in [0.5:1] can be a reasonable starting point for the calibration, and (2) with λ in that range our algorithm can improve the color consistency qualitatively and quantitatively even for the area not involved in the calibration. In this paper we used $\lambda = 0.75$ for all calibrations.

Table 1 Averaged color differences at corresponding points.

Original	Figure 5	Figure 9 (a)	Figure 9 (b)
15.60	8.532	9.635	11.98

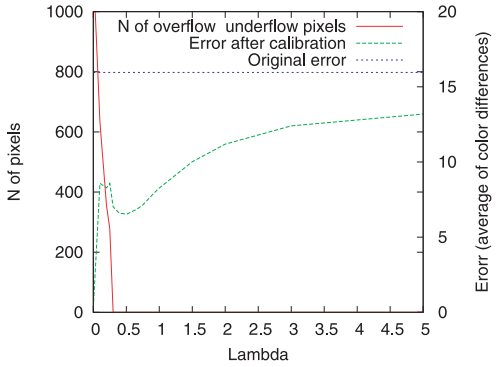
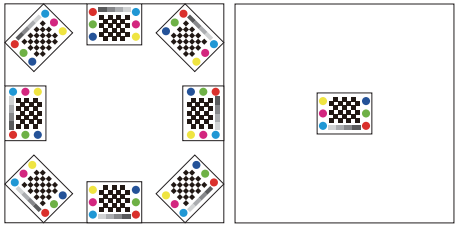


Fig. 12 Parameter λ and calibration error.



(a) Pattern C (b) Pattern D

Fig. 13 Reference objects for leave-one-out evaluation.

6.1.3 Comparison with a Conventional De-vignetting Method

To evaluate the vignetting parameters estimated by our method, we show our de-vignetting results without the imager response calibration and the result by

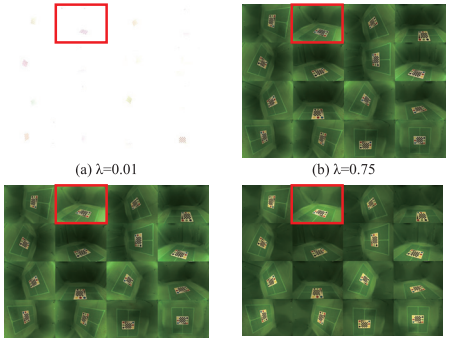


Fig. 14 Calibration results with various λ. Images with the red border indicate a camera corrected clearly.

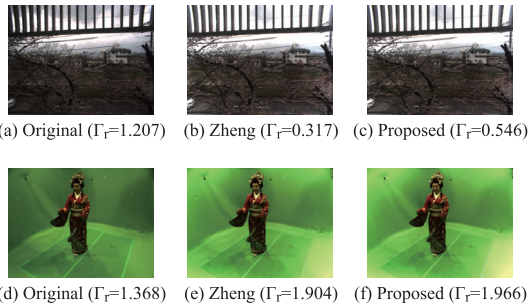


Fig. 15 De-vignetting results. The top row shows (a) the original, the result by (b) Zheng, et al. [11], and (c) ours for an outdoor scene. The bottom row shows (d) the original, the result by (e) Zheng, et al. and (f) ours for a studio scene.

Zheng, et al. [11]. Zheng, et al. use the same vignetting model (Eq. (19) of Ref. 11)) and estimate the parameters from a single natural image. We used the camera denoted by the red rectangle in Fig. 7 (Pointgrey Grasshopper and Kowa LM5JC1M lens).

The top row of **Fig. 15** shows (a) an original natural image, (b) the image de-vignetted by Zheng, et al., and (c) the image de-vignetted by our parameters. The de-vignetting parameters f and a for (b) is estimated from this single image by Zheng's method. We used the parameters f and a in the result of Fig. 7 for (c).

Table 2 Vignetting parameters.

	f	a
Zheng, et al.	1.286×10^3	2.62×10^{-6}
Proposed	1.159×10^3	7.83×10^{-8}

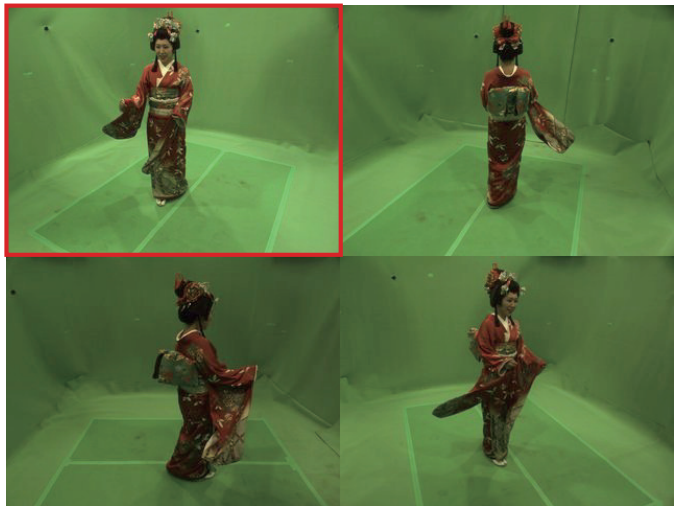


Fig. 16 Input images.

Ours and Zheng's parameters are described in **Table 2**. The Γ_r value in Fig. 15 is the asymmetry measure of radial gradient distribution in the image defined by Eq. (8) of Ref. 11). Zheng, et al. have revealed that de-vignetting can be done by minimizing this asymmetry measure if the image is taken in a natural lighting environment. In this result, we can observe that both Zheng's and our results reduce Γ_r, and they are successfully de-vignetted. Therefore we can conclude that the de-vignetting quality of our method is comparable to Zheng's method while our method does not rely on the asymmetry measure.

The bottom row of Fig. 15 shows the de-vignetting results of a multi-viewpoint studio image (the top left image of **Fig. 16**). Note that the image (f) is not equal to the top left image of **Fig. 17**. In Fig. 17, the de-vignetting process made the radial intensity fall-off be flat, and the gain calibration process made the global

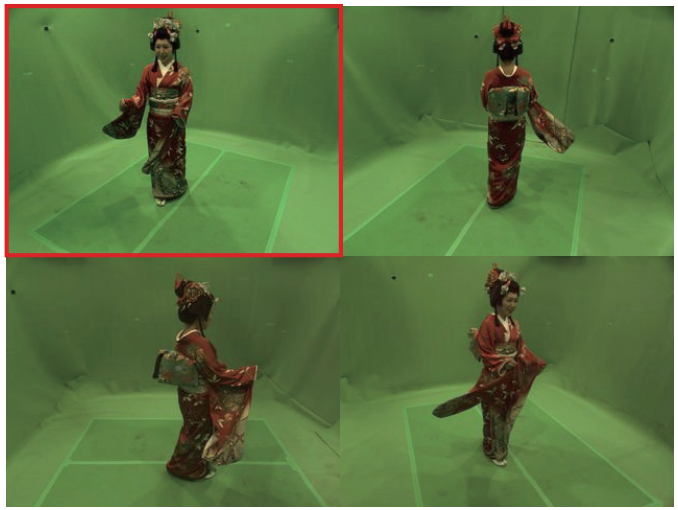


Fig. 17 Calibration results using parameters estimated with the reference object.

intensity lower. On the other hand, here we do not apply the imager response model in Eq. (7) in order to compare the de-vignetting part solely. In this result, we used the same parameters in Table 2. Because the de-vignetting parameter is independent to the scene, the parameter estimated by Zheng's method for the outdoor scene is valid for this studio scene as well by definition.

However the asymmetry measure Γ_r is not reduced both in (e) and (f). This means that de-vignetting cannot be achieved by minimizing this asymmetry measure based on "the natural image statistics" assumption. From these observations, we conclude that our algorithm can estimate de-vignetting parameters for near light sources environment while Zheng's method cannot work.

6.2 Evaluation Using Real Object

Then we evaluated our method using corresponding points estimated from a real object. Figure 16 shows the input multi-viewpoint images. **Figure 18** shows the 3D points computed from point correspondences estimated by the method described in Section 3. In this evaluation we obtained about 48,000 correspondences over 16 cameras. The processing time was approximately 10 minutes by an Intel Core-i7 860 PC.

Fig. 18 Obtained 3D points.

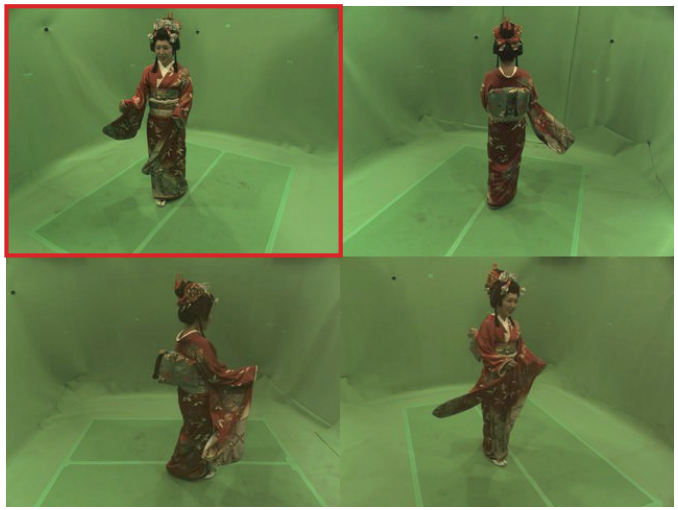


Fig. 19 Calibration results using points estimated from the real object.

Figure 19 shows the calibrated images using points estimated from the object itself (Fig. 18). Images calibrated with parameters estimated by the reference object (Fig. 5) are shown by Fig. 17 for comparison. By comparing the images marked by red borders in Figs. 16, 19 and 17, we can observe that the calibration result using the points from the real object is almost equal to the result with the

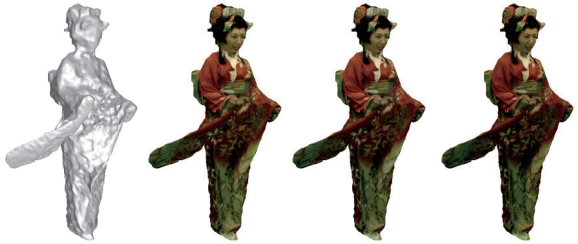


Fig. 20 Texture mapping results. (a) Reconstructed 3D shape [2], (b) Result with the original images, (c) Result with images calibrated by the reference object, (d) Result with images calibrated by the real object.

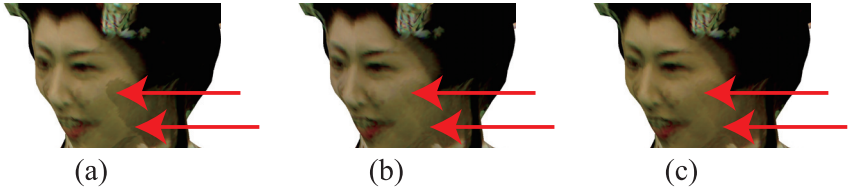


Fig. 21 Texture mapping results (close-ups). (a) Result with the original images, (b) Result with images calibrated by the reference object, (c) Result with images calibrated by the real object. Red arrows indicate boundaries of texture mapping where a texture-source camera switches to another.

reference object qualitatively.

Figures 20 and **21** show texture mapping results. We used the method in Ref. 2) to reconstruct the 3D shape, and generate its texture without any blending technique. While blending multiple camera images can improve the rendering quality [20], we do not use it in order to emphasize the color-calibration result. The red arrows in Fig. 21 indicate boundaries of mapped textures where a texture-source camera switches to another. By focusing on these areas, we can observe that the calibration using the points given by the real object (Fig. 21 (c)) produces smoother texture boundary when compared with the original (Fig. 21 (a)), and the quality is reasonably equal to that of the result using the reference object (Fig. 21 (b)). Hence we can conclude that our method can improve the color consistency across multiple cameras with corresponding points estimated automatically from the object itself.

Fig. 22 Camera arrangement and input images. (a) 12 XGA cameras. (b) Example frames taken by one of 12 cameras.

However, we can observe that calibrated images are still biased to green as the original images. This bias is originally caused by the reflections from the green background, and can be found in all camera images. Because our color calibration algorithm minimizes the color differences without knowing the "true" color, these global biases cannot be corrected by definition. This is a limitation of our method.

6.3 Calibration Using Multiple Frames

In this evaluation we show a failure case of calibration using features in a single frame, and then demonstrate that it can be solved by combining feature points in multiple frames. **Figure 22** (a) illustrates the studio setup of this experiment. We used 12 XGA cameras (Sony XCD-X710CR with Fujinon DF6HA-1B lens) running at 25 fps and captured a dance performance (Fig. 22 (b)).

Figure 23 shows the original images of frame 0 and images calibrated with feature points in this frame only. We can observe that the image of camera 2 is over de-vignetted. This is because we have no feature points in such areas in this frame. By contrast, the results using feature points from multiple frames do not include such an over de-vignetting (Figs. 23 (c) and (d)). For the results in Figs. 23 (c) and (d), we used three frames (0, 100 and 200) and 21 frames (0, 10, ..., 200) respectively. The processing time was approximately 2, 5, and 60 minutes for (b), (c), and (d) by an Intel Core-i7 860 PC respectively. We can conclude that (1) it is crucial to involve widely distributed feature points and (2) use of multiple frames can solve this problem.

In addition, **Fig. 24** shows calibrated images of another frame not used in the

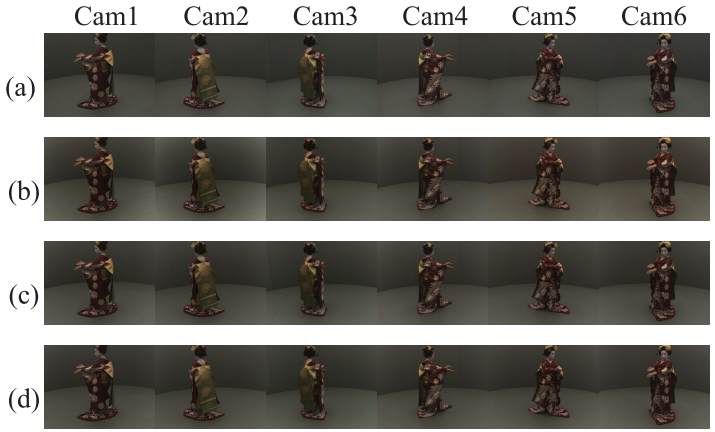


Fig. 23 Input and calibration results. (a) Original input images (Frame 0), (b) Calibrated images using features found in this frame, (c) Calibrated images using features found in three frames (Frame 0, 100, and 200), (d) Calibrated images using features found in 21 frames (Frame 0, 10, . . ., 200).

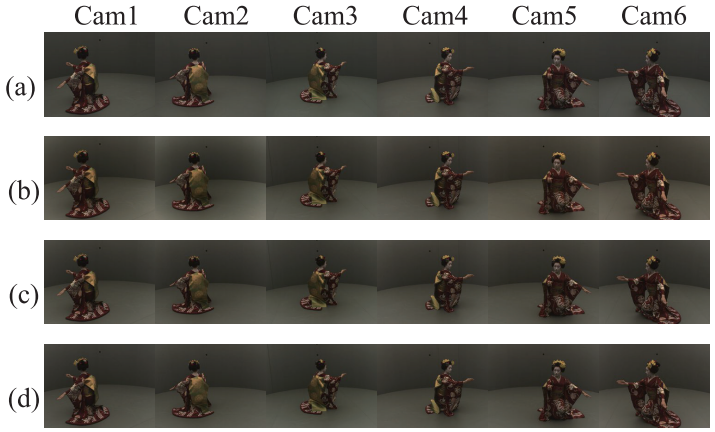


Fig. 24 Input and calibration results. (a) Original input images (Frame 900), (b) Calibrated images using the parameter of Fig. 23 (b), (c) Calibrated images using the parameter of Fig. 23 (c), (d) Calibrated images using the parameter of Fig. 23 (d).

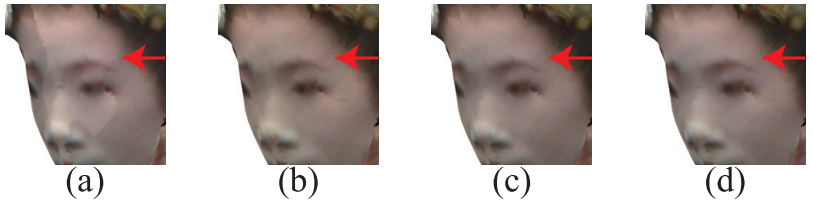


Fig. 25 Texture mapping results (frame 900). (a) Without color calibration, (b) Calibrated using the parameter of Fig. 23 (b), (c) Calibrated using the parameter of Fig. 23 (c), (d) Calibrated using the parameter of Fig. 23 (d).

parameter estimation process. Figure 24 (a) shows the original image of frame 900. Figures 24 (b), (c) and (d) show the results calibrated by the parameters of Figs. 23 (b), (c), and (d) respectively. We can observe that the estimated parameters can improve the color consistency of another frame, and therefore we conclude that the estimated parameters are reasonably frame-independent.

Besides, **Fig. 25** shows texture mapping results of frame 900. In this result, we can observe that the rendering quality is clearly improved even by the parameter estimated from a single frame (Fig. 23 (b)). This is because the position of the object is close to the position in frame 0, and therefore the calibration was valid at least for these areas. This is an important characteristic of our algorithm which minimizes the color differences on the 3D object surface, not the entire area of the image. From this observation, we believe that we can extend our approach for cameras changing their parameter (shutter, gain, hue, etc.) by combining frame-wise calibration with inter-frame color consistency constraints.

7. Conclusion

To improve the color consistency across cameras in sparse and convergent arrangement, we proposed a new method which estimates calibration points automatically from the object in the scene. Our approach does not require capture of reference objects, which is known to be a time-consuming and labor-intensive work. This also enables us to apply this method for images already taken by cameras without color calibrations, like images collected via Internet. In addition we can extend this technique for images taken with dynamic camera parameters (auto-exposure) typically used in an outdoor environment by introducing an ad-

ditional constraint on inter-frame color consistency.

On the other hand, the quantitative evaluation in Table 1 shows that our method cannot make the color differences across cameras to be zero exactly. This is because we have introduced the fidelity term D_0 into the objective function (Eq. (16)). The original idea was to prevent the optimization process from producing trivial solutions, but this term also prevents the optimization to make the color differences to be zero when the estimated ideal colors are not close to the original. As a result, the proposed method produces a set of parameters which minimize the color differences as long as the calibrated colors are not too far from the original colors. In addition to this point, our method (1) does not consider the non-Lambertian case, and (2) does not estimate the gamma correction in the imager by assuming machine vision cameras. These points are left for future work to achieve complete color consistency across multi-viewpoint non-machine vision cameras.

Acknowledgments This work was supported in part by MEXT of Japan under the GCOE program "Informatics Education and Research Center for Knowledge-Circulating Society", and JST-CREST project "Foundation of Technology Supporting the Creation of Digital Media Contents".

References

1) Starck, J., Hilton, A. and Miller, G.: Volumetric stereo with silhouette and feature constraints, *Proc. BMVC*, pp.1189–1198 (2006).
2) Tung, T., Nobuhara, S. and Matsuyama, T.: Simultaneous super-resolution and 3D video using graph-cuts, *Proc. CVPR*, pp.1–8 (2008).
3) Snavely, N., Seitz, S.M. and Szeliski, R.: Photo tourism: exploring photo collections in 3D, *Proc. SIGGRAPH*, pp.835–846 (2006).
4) Shao, F., Peng, Z. and Yang, Y.: Color correction for multi-view video based on background segmentation and dominant color extraction, *WSEAS Transactions on Computers*, Vol.7, No.11, pp.1838–1847 (2008).
5) Ilie, A. and Welch, G.: Ensuring Color Consistency across Multiple Cameras, *Proc. ICCV*, Vol.2, pp.1268–1275 (2005).
6) Joshi, N., Wilburn, B., Vaish, V., Levoy, M. and Horowitz, M.: Automatic Color Calibration for Large Camera Arrays, Technical report, UCSD CSE Technical Report CS2005–0821 (2005).
7) Porikli, F.M.: Inter-camera color calibration by correlation model function, *Proc. ICIP*, Vol.2, pp.133–136 (2003).
8) Goldman, D.B. and Chen, J.-H.: Vignette and Exposure Calibration and Compensation, *Proc. ICCV*, pp.899–906 (2005).
9) Litvinov, A. and Schechner, Y.Y.: Addressing Radiometric Nonidealities: A Unified Framework, *Proc. CVPR*, pp.52–59 (2005).
10) Kim, S.J. and Pollefeys, M.: Robust Radiometric Calibration and Vignetting Correction, *PAMI*, Vol.30, No.4, pp.562–576 (2008).
11) Zheng, Y., Yu, J., Kang, S., Lin, S. and Kambhamettu, C.: Single-image vignetting correction using radial gradient symmetry, *Proc. CVPR*, pp.1–8 (2008).
12) Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints, *IJCV*, Vol.60, No.2, pp.91–110 (2004).
13) Mikolajczyk, K. and Schmid, C.: Scale & Affine Invariant Interest Point Detectors, *IJCV*, Vol.60, No.1, pp.63–86 (2004).
14) Tola, E., Lepetit, V. and Fua, P.: A fast local descriptor for dense matching, *Proc. CVPR*, pp.1–8 (2008).
15) Furukawa, Y. and Ponce, J.: Accurate, Dense, and Robust Multi-View Stereopsis, *Proc. CVPR*, pp.1–8 (2007).
16) Asada, N., Amano, A. and Baba, M.: Photometric Calibration of Zoom Lens Systems, *Proc. ICPR*, Vol.1, pp.186–190 (1996).
17) Yu, W.: Practical anti-vignetting methods for digital cameras, *IEEE Transactions on Consumer Electronics*, Vol.50, No.4, pp.975–983 (2004).
18) Kang, S.B. and Weiss, R.S.: Can We Calibrate a Camera Using an Image of a Flat, Textureless Lambertian Surface?, *Proc. ECCV*, pp.640–653 (2000).
19) Hartley, R.I. and Zisserman, A.: *Multiple View Geometry in Computer Vision*, Cambridge University Press (2000).
20) Matsuyama, T., Wu, X., Takai, T. and Nobuhara, S.: Real-Time 3D Shape Reconstruction, Dynamic 3D Mesh Deformation and High Fidelity Visualization for 3D Video, *CVIU*, Vol.96, pp.393–434 (2004).

(Received November 10, 2009)
(Accepted August 9, 2010)
(Released November 10, 2010)

(Communicated by *Yasuhiro Mukaigawa*)

Shohei Nobuhara received his B.Sc. in Engeneering, M.Sc. and Ph.D. in Informatics from Kyoto University, Japan, in 2000, 2002, and 2005 respectively. From 2005 to 2007, he was a post-doctoral researcher at Kyoto University. Since 2007, he has been a research associate at Kyoto University. His research interest includes computer vision and 3D video. He is a member of IPSJ, IEICE, and IEEE.

Yuta Kimura received his B.Sc. in Engeneering from Kyoto University, Japan, in 2009. He is with Canon Corporation since 2009.

Takashi Matsuyama received his B.Eng., M.Eng., and D.Eng. degrees in electrical engineering from Kyoto University, Japan, in 1974, 1976, and 1980, respectively. He is currently a professor in the Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University. His research interests include knowledge-based image understanding, computer vision, 3D video, human-computer interaction, and i-Energy (advanced energy management). He wrote about 100 papers and books including two research monographs, A Structural Analysis of Complex Aerial Photographs, PLENUM, 1980 and SIGMA: A Knowledge-Based Aerial Image Understanding System, PLENUM, 1990. He won nine best paper awards from Japanese and international academic societies including the Marr Prize at ICCV'95. He is on the editorial board of Pattern Recognition Journal. He was given Fellowships from International Association for Pattern Recognition, Information Processing Society Japan, and Institute for Electronics, Information, and Communication Engineers Japan.