IPSJ Transactions on Computer Vision and Applications Vol.4 149–160 (Oct. 2012)

[DOI: 10.2197/ipsjtcva.4.149]

Research Paper

3D Face Reconstruction and Gaze Estimation from Multi-view Video using Symmetry Prior

Qun Shi[1,a)] Shohei Nobuhara[1] Takashi Matsuyama[1]

Received: October 31, 2011, Accepted: May 16, 2012, Released: October 19, 2012

Abstract: In this paper we propose a novel method that performs 3D face reconstruction, and non-constrained and non-contact gaze estimation on a moving object, whose head-pose can freely change, from multi-view video. The main idea is to first reconstruct the 3D face with high accuracy using symmetry prior. Then we generate a super-resolution virtual frontal face video from the estimated 3D face geometry and the original multi-view video. Finally a 3D eyeball model is introduced to estimate the three-dimensional gaze direction from the virtual frontal face video. Experiments with real data illustrate the effectiveness of our method.

Keywords: 3D shape reconstruction, gaze estimation, multi-view video, symmetry prior, super-resolution

1. Introduction

This paper is aimed at presenting a novel method to estimate the three-dimensional gaze direction from multi-view videos. In the literature of accurate 3D gaze estimation from video, conventional methods assume to have frontal face video of the object as their inputs [1]. Such methods are known to work robustly in practice [2], but they strictly limit the object's head motion within a very small range. Instead, we propose a "virtual frontal face video synthesis" approach that generates a frontal face video from regular multi-view videos. This approach allows the object's face to move freely in the scene, and therefore realizes a non-contact and non-constrained gaze sensing.

The ideas behind this method are as follows. Generally the accuracy of the reconstructed 3D shape data is limited due to errors in the calibration and shape reconstruction processes, which could mislead the gaze estimation and/or decrease its accuracy. Fortunately, the 3D face surface is rather flat, which allows many cameras to observe it, and moreover, it has symmetric properties in both 3D shape and surface texture. Thus a super-resolution technique with symmetry prior can be applied to increase the 3D shape accuracy and the image resolution, making full use of original multi-view images.

The overall processing scheme of the proposed method is as follows. As is shown in **Fig. 1**, given a sequence of 3D mesh data and corresponding multi-view video data, we first extract 2D face regions in multi-view images to estimate a rough 3D face surface area in each 3D mesh (Fig. 1 II and III). Then we estimate the symmetry plane of the 3D face surface area by: (1) first extracting 3D feature points in the estimated 3D face surface area and then, (2) generating the symmetry plane by evaluating symmetric properties among the feature points (Fig. 1 IV and V). Next, we reconstruct an accurate and high resolution frontal face surface by applying a super-resolution 3D shape reconstruction technique with the symmetry prior (Fig. 1 VI). Then a virtual frontal face image with super-resolution can be generated (Fig. 1 VII). Finally, we estimate the 3D gaze from the virtual frontal face image using a 3D eyeball model (Fig. 1 VIII).

The rest of this paper is organized as follows. We first review related studies to clarify the contribution of this work in Section 2. We introduce our 3D face surface reconstruction method in Section 3, our virtual frontal face image synthesis algorithm in Section 4, and 3D gaze estimation algorithm in Section 5. We evaluate our method with real data in Section 6. Finally, we summarize the proposed method in Section 7.

2. Related Work

2.1 3D Shape Reconstruction from Multi-View Video

Nowadays one popular way of full 3D shape (not 2.5D) reconstruction integrates both shape-from-silhouette [3], [4] and shape-from-stereo [5], [6] techniques. Shape-from-silhouette robustly estimates the rough object shape as a visual hull, and shape-from-stereo refines it if the surface is well-textured. For human faces, however, shape-from-stereo cannot perform well since most of the surface area is poorly-textured. To solve this problem, we introduce a symmetry prior constraint in our 3D shape reconstruction algorithm based on a mesh-deformation. While some studies have proposed mesh-deformation based algorithms which integrate shape constraints, such as smoothness or curvature, they are given as local constraints defined on each vertex and its proximity. On the other hand, our symmetry prior performs as a global constraint over the entire mesh surface. To the best of our knowledge, this is the first for full 3D shape reconstruction from multi-view images with such a global shape prior.

[1] Kyoto University, Kyoto 606–8321, Japan
[a)] seki@vision.kuee.kyoto-u.ac.jp

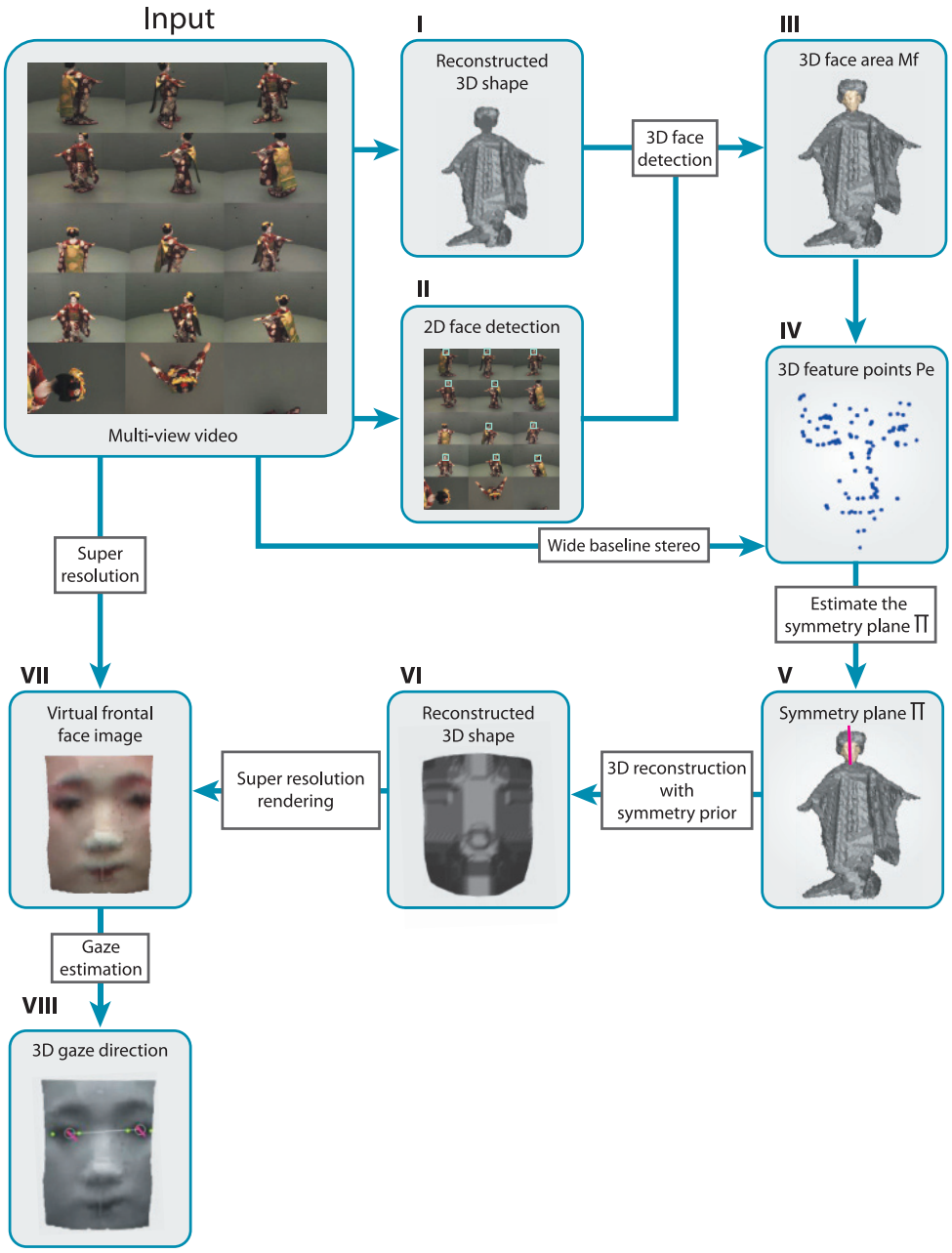


Fig. 1 Computational processes for 3D gaze estimation.

2.2 Super-resolution Using Multi-view Videos

Super-resolution techniques for estimated 3D shape from multi-view videos can be categorized into two groups: view-independent and view-dependent. The first group generates a super-resolution texture of the 3D surface [7]. The generated texture is optimized over the object surface, and is suited for view-independent rendering. The second group generates a super-resolution rendering of the object for a specified viewpoint [8]. The generated image is optimized for the viewpoint, and therefore is well suited for our virtual frontal face image synthesis. Based on this understanding, we employ the view-dependent approach which can generate an optimized super-resolution image of the object 3D face.

2.3 Gaze Estimation from Frontal Face Images

In gaze estimation literature, there exist a lot of studies using 2D frontal face images. Yamazoe et al. [9] proposed to track 2D eye features from images captured by a single camera to estimate the horizontal and vertical gaze angles in 3D space. The use of Active Appearance Models (AAM) [10] and a 3D eyeball model has been proposed by Ishikawa et al. [11]. In Guestrin et al.'s work [12], a single camera with multiple calibrated light sources are used for 3D gaze estimation. And Matsumoto et al. [13] proposed to use a stereo camera to estimate the 3D eye position and 3D visual axis. Besides, by combining image saliency with a 3D eye model, Chen et al. [14] proposed a probabilistic gaze estimation method that requires no active personal calibration. In addition, the research of Weigle et al. [2] verified the effectiveness of a commercial eye-gaze tracker, Tobii. While these works have realized effective and robust gaze estimation, they all suffer from the drawback that the head motion of the object is strictly limited within a small range, making it impossible to estimate the gaze

direction from a freely moving object. The main contribution of our work is the proposition of a method that can generate virtual frontal face images and perform gaze estimation on freely moving objects. Once we obtain a virtual front face image of the object from multi-view videos as we have introduced above, we utilize a conventional method which generates 3D gaze direction from an image with an extension in computing global 3D gaze direction [11]. In conventional 3D gaze direction estimation from a 2D image, calibration is required to map apparent gaze directions to those in a world coordinate system. Thanks to the fully-calibrated multi-view camera environment, we can easily convert apparent gaze directions into the global world system.

3. 3D Face Surface Reconstruction Using Symmetry Prior

In this section we present the super-resolution 3D shape reconstruction algorithm using symmetry prior from the 3D mesh and corresponding multi-view images. It consists of (1) 3D face area detection, (2) symmetry plane estimation and (3) 3D face surface reconstruction in super-resolution. The algorithm processes frames one-by-one sequentially.

3.1 3D Face Area Detection

First we propose an algorithm to detect the 3D positions and directions of the object's face from multi-view videos. The basic idea is to use a 3D mesh as a *voting space* for accumulating partial evidence produced by applying an ordinary 2D face detector to each of the multi-view images. The evidence accumulation enables us to (1) eliminate false-positive face detections in 2D images and (2) localize an accurate 3D face area on the 3D mesh.

Let M denote a 3D mesh of an object and $I_i (i = 1, \ldots, N)$ a set of corresponding multi-view images captured by cameras $c_i (i = 1, \ldots, N)$. The face area detection algorithm (Fig. 1 II and III) is defined as follows. Note that in what follows, Step X denotes the process X illustrated in Fig. 1.

First the algorithm detects a set of 2D face candidate regions F_i by applying a conventional 2D face detector to each I_i. The blue rectangles in **Fig. 2** show F_i for each image. It should be noted that F_i may include false-positive face areas due to texture patterns which accidentally look like a human face. Then all F_is are mapped onto M for evidence accumulation.

Step II Apply Viola-and-Jones face detector [15] to each image $I_i (i = 1, \ldots, N)$ to obtain a group of face candidate regions $F_i = \{f_{ij} | f_{ij} \in I_i, j = 1, \ldots, n_i\}$, where n_i denotes the number of face candidate regions in F_i.

Step III-1 Let $M = \{V, E\}$ denote a 3D mesh consisting of a vertex set V and an edge set E. For each vertex $v \in V$, compute a per-vertex "faceness" score $L(v)$ by the following method:

Step III-1-1 For each v, let $L(v) = 0$.

Step III-1-2 For each camera c_i, let v_i denote the projection of vertex v on image I_i. If v_i falls in F_i, then let $L(v) = L(v) + 1$.

Step III-2 Compute a set of vertices $V_L = \{v | L(v) > 0\}$, and partition it into disjoint subgroups of connected vertices $S = \{s_i | s_1 \cup s_2 \cup \ldots s_n = V_L, s_j \cap s_k = \emptyset\ (j \neq k)$, all vertices in s_i are connected. $\}$. Here n denotes the number of the subgroups.

Fig. 2 2D face detection in multi-view images. Blue rectangles denote the detected 2D face candidate regions.

Fig. 3 Detected 3D face area M_f painted in skin color.

Step III-3 For each vertex group s_i in S, calculate the average of $L(v)$ by:

$$\bar{L}_i(v) = \frac{\sum_{v \in s_i} L(v)}{N(s_i)}, \tag{1}$$

where $N(s_i)$ denotes the number of vertices in s_i.

Step III-4 Find the s_i with the largest $\bar{L}(v)$ and denote it by V_f.

$$V_f = \arg\max_{s_i} \bar{L}_i(v) \tag{2}$$

Return the sub-mesh $M_f = \{V_f, E_f\}$ as the 3D face area.

Figure 3 shows the detected 3D face area M_f.

3.2 Symmetry Plane Estimation

The assumption that human faces have symmetric properties in both 3D shape and surface texture allows us to reconstruct a more accurate 3D face surface than M_f and hence generate a higher resolution frontal face image than captured images.

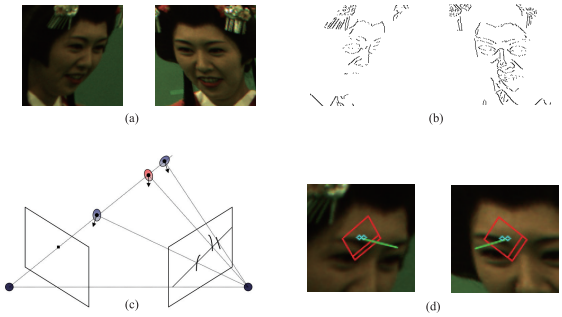


Fig. 4 Matching based on edge features. (a) rectified images, (b) edge features crossing epipolar lines, (c) texture similarity computation with normal direction optimization, (d) an example of a matched pair. In (d), the red rectangles illustrate the windows used to compute the texture similarity, the green lines the surface normals, and the blue circles the endpoints of the edge features.

The symmetry plane detection from M_f consists of two processes: (1) detect 3D feature points P_e on M_f (Fig. 1 IV) and (2) apply RANSAC [16] to estimate the symmetry plane based on P_e (Fig. 1 V).

3.2.1 3D Feature Points Extraction

In order to find the symmetry plane that divides M_f into two symmetric parts, we first extract 3D feature points P_e on the local object surface specified by M_f. To avoid possible artifacts introduced by the texture generation, we apply a stereo-based edge feature detection method to the multi-view images as illustrated in **Fig. 4**. That is, we establish sparse but reliable 2D-to-2D correspondences to obtain 3D feature points by triangulation [17]. This algorithm is based on the wide-baseline stereo by Furukawa et al. [18] and augmented by a bi-directional uniqueness examination to improve the accuracy and robustness of the matching.

Step IV-1 Project M_f back onto the multi-view images to localize 2D face regions, respectively. Let c and c' denote a pair of cameras whose images include well captured 2D face regions. Rectify the images captured by c and c' for stereo matching (Fig. 4 (a)) and extract edge features from the 2D face regions in the rectified images.

Step IV-2 Eliminate edge features which do not cross the epipolar lines. Let I_E and I'_E denote the resultant edge feature images (Fig. 4 (b)). Let e denote a point on an edge feature in I_E, l' the corresponding epipolar line in I'_E, and $E' = \{e'_j | j = 1, \ldots, n\}$ the points on the edge features in I'_E intersecting with l'.

Step IV-3 Compute the texture similarity between e and $e'_j \in E'$ using the normal direction optimization [18] with the ZNCC photo consistency evaluation. (Fig. 4 (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 and only if the similarity between them is significantly better than the second best pair. Otherwise we reject this pair and leave e without correspondence to avoid ambiguous matching.

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

Step IV-5 By iterating the steps from IV-2 to IV-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 $\langle p^i_c, p^i_{c'} \rangle$ denotes a corresponding point pair and $n_{c,c'}$ the number of obtained correspondences.

Step IV-6 By collecting $P_{c,c'}$ computed from all possible pairs of cameras that can observe the face area M_f, we can compute a set of 3D feature points, P_e, from a set of matching 2D point pairs.

3.2.2 Symmetry Plane Estimation Using 3D Feature Points

Having computed the reliable 3D feature point set $P_e = \{p_i\}$ $(i = 1, \ldots, N)$ in Section 3.2.1, we then estimate the symmetry plane π from P_e as follows (Fig. 1 V). The idea is to generate a candidate symmetry plane π and compare the texture pattern around p_i with that of its symmetric position with respect to π. If π is a valid symmetry plane, then the textures should be reasonably similar.

Step V-1 Randomly pick two points p_i, p_j $(i \neq j) \in P_e$, and repeat the following processing for $K \leq N(N-1)/2$ times.

Step V-1-1 Compute the symmetry plane π_{ij} that makes p_i and p_j in the symmetric position.

Step V-1-2 Based on the hypothesized symmetry plane π_{ij} we can compute the symmetric position for each of the other $N-2$ points. Let $\breve{p}_k$ denote the symmetric position of p_k $(k \neq i, j)$. Then we compare the textures at p_k and $\breve{p}_k$. First we generate two $L \times L$ grids centered at p_k and $\breve{p}_k$ in the 3D space. Note that these two grids lie on the planes that are perpendicular to the hypothesized symmetric plane, and the distance between neighboring grid points is d, which is a variable free to change according to the size of the 3D object. Since the 3D position of each grid point is computable, let p^{mn}_k, $\breve{p}^{mn}_k (0 \leq m \leq L, 0 \leq n \leq L)$ denote the grid points on the grids centered at p_k and $\breve{p}_k$. And let $Col(p^{mn}_k)$ and $Col(\breve{p}^{mn}_k)$ denote the RGB color vectors of the grid points p^{mn}_k and $\breve{p}^{mn}_k$ respectively, which are computed from the images by their best-observing cameras. Here we use M_f as the shape proxy for the state-based visibility evaluation [19]. Then the texture dissimilarity between p_k and $\breve{p}_k$, d_{p_k}, is computed as Sum-of-Absolute-Difference:

$$d_{p_k} = \sum_{0 \leq m \leq L, 0 \leq n \leq L} \left| Col(p^{mn}_k) - Col(\breve{p}^{mn}_k) \right|. \quad (3)$$

Note that if either p^{mn}_k or $\breve{p}^{mn}_k$ is located outside the estimated face area, the point pair is considered as an outlier and a fixed value $diff$ is set to $\left| Col(p^{mn}_k) - Col(\breve{p}^{mn}_k) \right|$. By computing d_{p_k} for all p_k $(k \neq i, j)$, we can evaluate the goodness of π_{ij} by

$$d_{ij} = \sum_{p_k \in P_e \setminus p_i, p_j} d_{p_k}. \quad (4)$$

Step V-2 Select the symmetry plane π_{ij} having the smallest $d_{i,j}$ as the symmetry plane π.

In experiments, we used $L = 4$ and $d = 5$ mm. The number of 3D feature points, N, was about a few hundred, while changing from frame to frame.

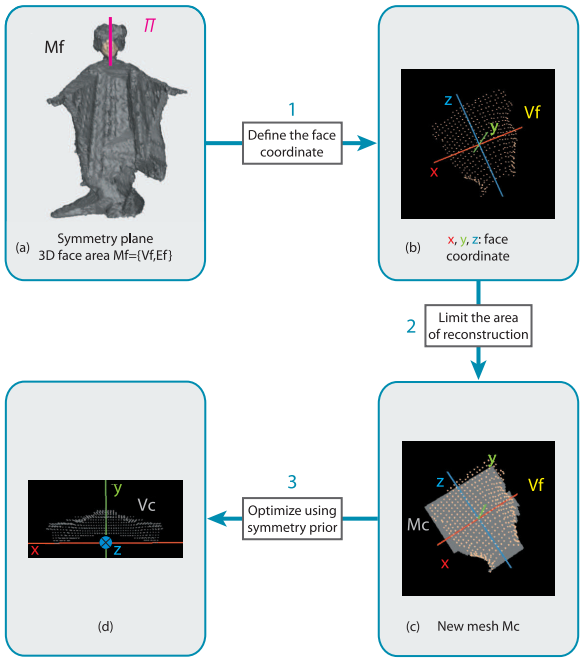


Fig. 5 3D face shape reconstruction using symmetry prior.

3.3 3D Shape Reconstruction Using Symmetry Prior

As is mentioned in Section 1, the shape reconstruction process in Fig. 1 I estimates the 3D object surface geometry without any specific knowledge nor object model, which results in the limited reconstruction accuracy and the introduction of errors. By contrast, the 3D shape reconstruction algorithm in this section utilizes the knowledge of symmetric properties of the human face to attain more accurate and higher resolution 3D shape reconstruction (Fig. 1 VI). The algorithm is similar to the mesh-deformation algorithm proposed by Nobuhara et al. [20], but employs the symmetry constraint in deforming the mesh. This section first describes how we model the 3D face surface by a mesh model, and then introduces how we can utilize the symmetry prior as a constraint on the mesh deformation.

The processes so far described have generated the 3D face area $M_f = \{V_f, E_f\}$ as a sub-area of the original 3D mesh surface M and estimated its symmetry plane π (**Fig. 5** (a)). With this symmetry plane, we first define the 3D face coordinate system as illustrated in Fig. 5 (b): define the origin by the centroid of V_f and place the coordinate axes so that the symmetry plane π is aligned with the $x = 0$ plane, the X-axis is defined by the normal vector of π, and the Z-axis by the principal axis of the point distribution of V_f on π. The Y-axis is computed by the cross-product of the other axes.

Then we generate a new mesh $M_c = \{V_c, E_c\}$ to model the higher resolution 3D face surface: project M_f onto the $y = 0$ plane and define a bounded regular mesh M_c on the 2D projected region. The gray area in Fig. 5 (c) illustrates M_c. That is, V_c and E_c denote the set of grid points and edges in this projected region, respectively. Note that the sampling pitch by the regular grid can be designed to increase the spatial resolution.

With this modeling, the 3D face surface reconstruction problem is transformed to that of finding the appropriate y value of each regular grid point in V_c (Fig. 5 (d)). Here the technical problems to be solved are (1) how we can introduce the symmetry constraint into the mesh deformation and (2) how we can find the optimal y values for V_c.

First, we represent the symmetry prior by

$$y = f(x, z) = f(-x, z), \tag{5}$$

where the function $f(x, z)$ returns the y value of the grid point at (x, z). Then, introduce the following discrete representation of y values:

$$y = \alpha i, \tag{6}$$

where i denotes an integer within a certain range, and α specifies the resolution of possible y values.

This discrete modeling allows us to formalize the shape reconstruction problem as a multi-labeling problem. That is, we can formulate the shape reconstruction with the symmetry prior as the minimization of the following objective function:

$$\mathscr{E}(M_c) = \sum_{v \in V_c, v_x \geq 0} \mathscr{E}_p(i_v) + \sum_{(u,v) \in E_c, u_x, v_x \geq 0} \mathscr{E}_c(i_u, i_v), \tag{7}$$

where v_x and u_x denote the x coordinate values of v and $u \in V_c$ respectively, and i_v and i_u integer labels to specify y values at v and u respectively. $\mathscr{E}_p(i_v)$ denotes the photo-consistency evaluation function at v and its symmetric position $\breve{v}$. That is,

$$\begin{aligned} \mathscr{E}_p(i_v) &= \rho(v_x, \alpha i_v, v_z) + \rho(\breve{v}_x, \alpha i_{\breve{v}}, \breve{v}_z) \\ &= \rho(v_x, \alpha i_v, v_z) + \rho(-v_x, \alpha i_v, v_z), \end{aligned} \tag{8}$$

where $\rho()$ denotes the photo-consistency evaluation function based on the state-based visibility with M as the shape proxy [19]. $\mathscr{E}_c(i_u, i_v)$ evaluates the smoothness in the y direction between a pair of connected grid points v and u:

$$\mathscr{E}_c(i_u, i_v) = \kappa |\alpha i_u - \alpha i_v| \tag{9}$$

where κ is a weighting factor to balance the photo-consistency and smoothness terms. This formalization forces the mesh deformation to satisfy the symmetry constraint defined by Eq. (5). We solve this minimization problem by belief-propagation [21], and obtain the 3D face surface satisfying both the photo-consistency and the symmetry constraint simultaneously.

In experiments, we used $\kappa = 1.0$, $\alpha = 1$ mm, and 2.5 mm grid resolution for M_c. Note that the original 3D mesh resolution, i.e., the average distance between adjacent vertices of M_f was about 4.7 mm.

4. Virtual Frontal Face Image Synthesis

With the optimized M_c, the virtual frontal view of M_c is generated for gaze estimation (Fig. 1 VII): (1) locate a virtual camera with focal length f at $(0, P_{cam}, 0)$ and align its view direction at $(0, 0, 0)$ in the 3D face coordinate system defined in Section 3.3, and then (2) generate the virtual frontal face image by rendering M_c from the virtual camera by the super-resolution technique proposed by Tung et al. [8]:

Step VII-1 Set a high-resolution pixel grid on the image plane of the virtual camera.

Step VII-2 Project each pixel of the original multi-view images,

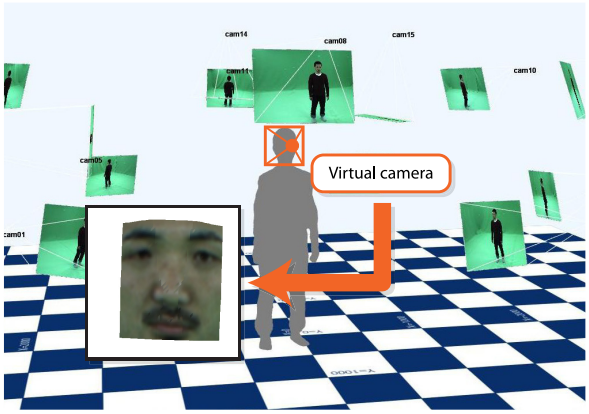


Fig. 6 Virtual frontal face image synthesis.

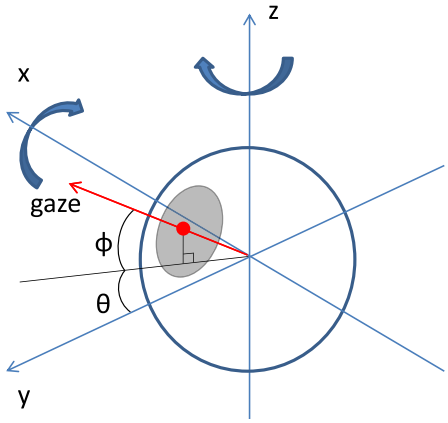


Fig. 7 3D eyeball model.

say source pixels, onto the pixel grid via M_c. That is, back project the source pixels onto M_c first, and then project the points on M_c to the pixel grid of the virtual camera. In this process we choose the nearest grid point as the final projection point of each source pixel. In addition we ignore source pixels if their projections are occluded by M.

Step VII-3 For each grid point with source pixel projections, compute its color by averaging associated source pixel colors. Otherwise, interpolate the grid point color using colors of its neighbors.

Figure 6 shows a synthesized virtual front face image, where the image resolution is increased by the super-resolution rendering process.

In experiments, we used $f = 430\,\text{mm}$, $P_{cam} = 500\,\text{mm}$, and the virtual face image plane of $160\,\text{mm} \times 160\,\text{mm}$ sampled with 400×400 pixels. Considering that the average of y values in a 3D face area is about 15 mm, the size of the virtual image pixel projected on the 3D face surface is about $0.45\,\text{mm} \times 0.45\,\text{mm}$, which is much higher than 4.7 mm, the average distance between adjacent vertices in the original 3D mesh. Note that the resolution of the original multi-view images is higher than that of the original 3D mesh. It was estimated at most at about 1.5 mm on the face area. That is, the super-resolution attained about three times higher resolution than the original images.

5. Gaze Estimation Using 3D Eyeball Model

For the last stage, we propose to introduce a 3D eyeball model to estimate the object's gaze direction (Fig. 1 VIII). **Figure 7** illustrates the structure of the model. The red arrow indicates the 3D gaze direction, and θ and φ denote the horizontal and vertical rotation angles of the eyeball, respectively. This model is designed based on the following three assumptions:

(1) The eyeball is fixed inside the eye socket and it can rotate horizontally and vertically around the eyeball center.

(2) The gaze direction is defined by the 3D vector pointing from the eyeball center to the iris center.

(3) The radius of the eyeball is equal to the diameter of the iris. This assumption is made based on medical statics data.

To apply this model to the 3D gaze estimation, the eyeball model of the object should be estimated first by the following off-line process:

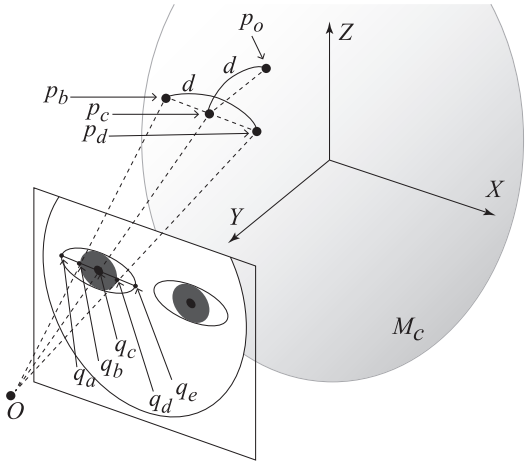


Fig. 8 Eyeball center position estimation.

Step VIII-1 Collect virtual frontal face images in which eyes look straight forward by hand.

Step VIII-2 For each image, detect the following eye feature points (**Fig. 8**) for each eye: 2D eye corners, q_a and q_e, 2D iris center, q_c, and the intersecting points between the iris border and the eye corner line connecting q_a and q_e, q_b and q_d. The eye corners are located by the AAM [10], and the iris is detected by applying the method of Kawaguchi et al. [22]. Note that all feature points for the right eye illustrated in Fig. 8 are mirrored with respect to the symmetry plane to represent those for the left eye.

Step VIII-3 For each eye, let d denote the average 3D diameter of the iris, and consequently the eyeball radius. The 3D diameter of the iris is defined by the 3D distance between p_b and p_d on the face surface M_c, which are obtained by back-projecting q_b and q_d onto M_c respectively.

Step VIII-4 For each eye, compute the average 2D relative position t of the iris center q_c with respect to the eye corners q_a and q_e. That is, t denotes the weighting parameter to represent q_c by the weighted average of q_a and q_e: $q_c = (1-t)q_a + tq_e$ where $t = |q_c - q_a| \,/\, |q_e - q_a|$.

This process estimates the eye model parameters for the left and right eyes respectively: d_{left} and t_{left}, and d_{right} and t_{right}. In what follows, we eliminate the suffix for simplicity.

With the eye ball model parameters d and t, compute the 3D gaze directions of the left and right eyes from each 3D video

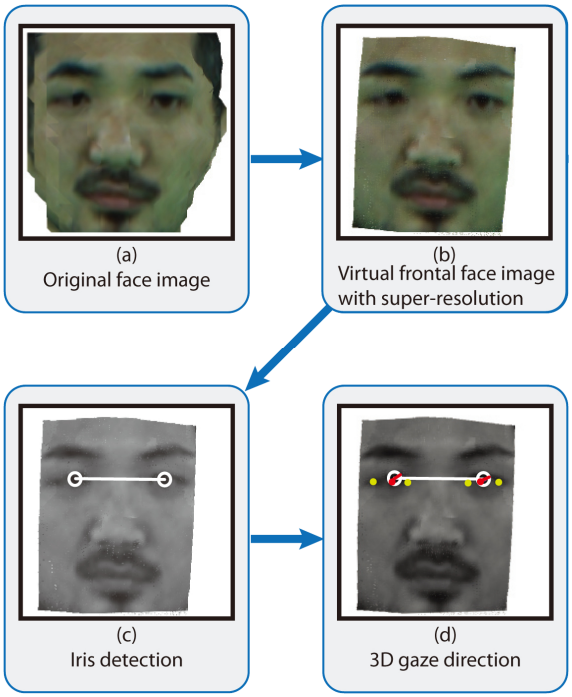


Fig. 9 Gaze estimation process. (a) original face image generated based on the original 3D mesh M (b) virtual super-resolution frontal face image generated based on the reconstructed face surface M_c (c) detected irises (circles) (d) estimated eye corners (green dots) and gaze directions (red lines).

frame by the following process (Fig. 8 and **Fig. 9**). Note that all 3D points as well as the virtual frontal face image in the 3D gaze estimation below are represented in the face coordinated system defined in Section 3.3 and Fig. 5 (b), which is dynamically defined depending on the 3D face position and direction in each 3D video frame.

Step VIII-5 Apply the following process to the left and right eyes, respectively.

Step VIII-6 Detect 2D eye corners q_a and q_e, and the iris center q_c from the synthesized frontal face image.

Step VIII-7 Compute the 3D iris center position p_c by back-projecting q_c onto M_c.

Step VIII-8 Compute the 3D eyeball center p_o by

$$p_o = p_{\tilde{c}} + (0, -d, 0)^\top. \tag{10}$$

Here $p_{\tilde{c}}$ denotes the back-projection of $q_{\tilde{c}} = (1-t)q_a + tq_e$ onto M_c, where $q_{\tilde{c}}$ represents the 2D position of the assumed iris center if the eye were looking straight forward.

Step VIII-9 Finally the 3D gaze direction is given as the line passing through p_o and p_c.

6. Performance Evaluation

In this section we evaluate the performance of the proposed method with real data.

6.1 Shape Reconstruction Using Symmetry Prior

First, we present the analysis on how the accuracy of the reconstructed 3D face shape is improved by introducing the symmetry prior.

Fig. 10 Input multi-view data A.

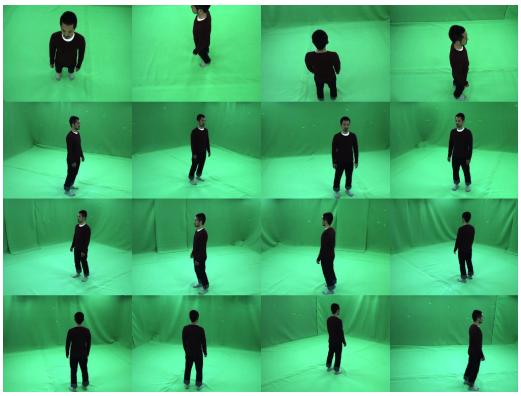


Fig. 11 Input multi-view data B.

6.1.1 Experiment Setup

For this evaluation we used two sets of data in doing the experiment. Data A (**Fig. 10**) is captured by 15 calibrated UXGA cameras running at 25 HZ with 1 msec shutter speed, while Data B (**Fig. 11**) is captured by 16 calibrated UXGA cameras running at 25 Hz with 1 msec shutter speed.

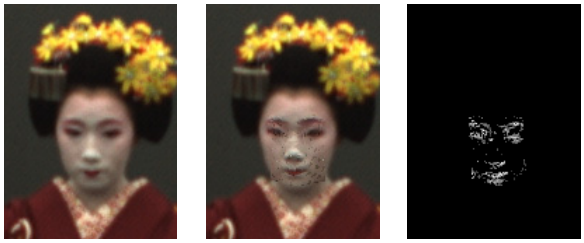
6.1.2 Evaluation Method

We measure the contribution of the symmetry prior to the reconstruction accuracy by means of leave-one-out experiments. We keep one camera c_f for evaluation, and use the other 15 cameras to render the face image viewed from camera c_f by the rendering algorithm described in Section 4. Note that the size and resolution of the rendered image is adjusted to coincide with that of the image captured by c_f. Let I'_f denote the rendered image. Then we compute the mean-squared-error between the rendered image I'_f and the originally captured image I_f:

$$\mathrm{MSE} = \frac{1}{N} \sum_{(x,y)\in I_f} (I_f(x,y) - I'_f(x,y))^2, \tag{11}$$

where N is the total number of effective pixels in I'_f.

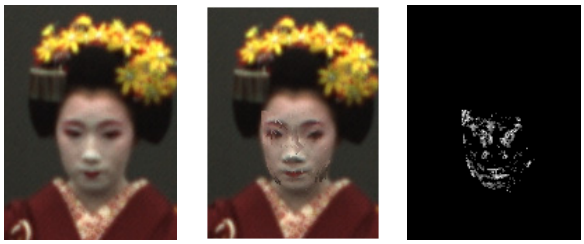
In this experiment we compute I'_f in two ways as follows: (1) the virtual view image with the original 3D shape. (2) the vir-



(a) Original captured image. (b) Virtual frontal face image. (c) Difference image.

(d) Enlarged Original captured image. (e) Enlarged Virtual frontal face image. (f) Enlarged Difference image.

Fig. 12 Difference image with the proposed method.



(a) Original captured image. (b) Virtual frontal face image. (c) Difference image.

(d) Enlarged Original captured image. (e) Enlarged Virtual frontal face image. (f) Enlarged Difference image.

Fig. 13 Difference image with the original 3D shape.

tual view image with the reconstructed 3D shape using symmetry prior. By comparing these two types of I'_f with the original image I separately, we can comprehensively evaluate the effect of introducing the symmetry prior.

6.1.3 Experiment Results

Figures 12 and **13** illustrate the difference images between the original captured image and the synthesized virtual frontal face image, generated with the original 3D shape and the reconstructed 3D shape using symmetry prior, respectively. In Fig. 12, the synthesized face image in the middle is less blurred than the original captured image, and the differences mainly occur on the edges, proving that the synthesized virtual frontal face image with the proposed method has higher resolution than the original captured image. As for the one with the original 3D face shape, obvious differences appear on the cheeks as well as the edges. **Table 1** illustrates that the mean-squared-error MSE of the proposed method is smaller than using the original 3D face shape.

Table 1 MSE with the original captured image.

	MSE	Effective Pixels
Proposed method	10.23	1,900
Original 3D shape	11.07	2,223

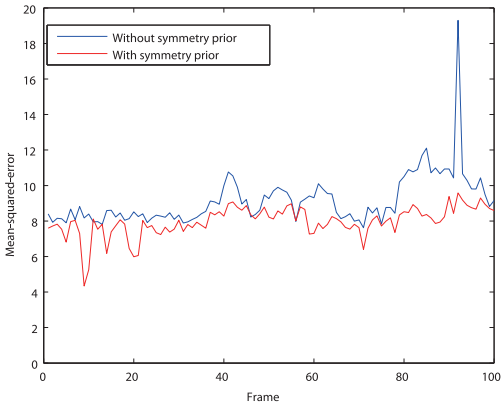


Fig. 14 Mean-squared-errors between the synthesized and the original images.

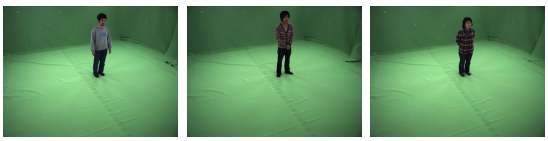


Fig. 15 Multi-view input data with three different people.

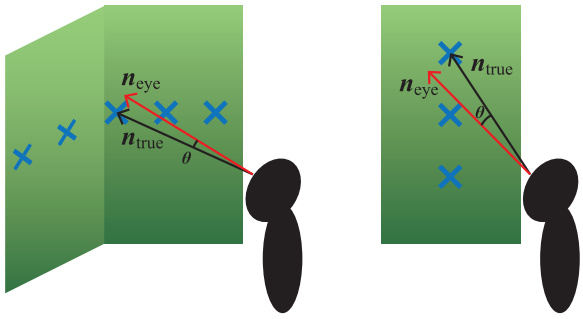


Fig. 16 Gaze estimation error evaluation.

Figure 14 shows the mean-squared-errors over 100 continuously captured frames using Data B. It can be observed that the symmetry prior contributes to improve the fidelity of rendering and hence improve the reconstruction accuracy.

6.2 Gaze Estimation

To prove the effectiveness of our gaze estimation method, we prepared the input multi-view videos captured with three different people for evaluation (**Fig. 15**). We have also conducted the experiment with an effective commercial gaze tracking device Tobii X120 Gaze Tracker in the same environment for comparison.

6.2.1 Experiment Setup

In order to quantitatively evaluate the accuracy of the gaze estimation processing, we designed our experiment as follows. **Figure 16** illustrates the experimental environments. A human subject stands at about 2.5 m away from the wall and looks at (1) horizontally aligned markers one by one, and (2) vertically aligned markers one by one (Fig. 16 left and right respectively). For each marker, its 3D position p_m in the object oriented coordinate system is measured manually. **Figure 17** is the template used for iris

detection based on the method of Kawaguchi et al. [22], with a size of 300×122 pixels.

As is shown in **Fig. 18**, experiments using a Tobii X120 Eye Tracker are conducted as well in the same environment. The subject stands at the same position as in the experiment with multi-view cameras. A Tobii X120 Eye Tracker is set in front of the subject for iris and gaze tracking, and a projector is used to project the marker image onto the wall for the subject to look at. Since the positions of the two outer markers in the horizontal direction (Fig. 16 left) are out of the effective region of Tobii, we only project the other five markers onto the wall. It should be noted that the marker image is well designed to project each marker into the same position as the corresponding one's position in the multi-view camera experiment. Besides, we manually set two markers onto the positions of the two outer ones in the horizontal direction (one of them is highlighted with a red circle in Fig. 18), which could be used to estimate the performance of Tobii when the gazing position is outside its effective region. **Table 2** illustrates the technical specifications of the Tobii X120 Eye Tracker being used in our experiment.

6.2.2 Evaluation Method

In the multi-view video data, we first selected those video frames where the subject was stably looking at each marker. Then, for each selected video frame, apply the above mentioned gaze estimation processes from Step I to Step VIII to obtain the estimated 3D gazing vector $\boldsymbol{n}_{\text{eye}}$. Note that the gaze estimation is conducted for the left and right eyes independently, meaning that we have $\boldsymbol{n}_{\text{eye}}$ for each eye. The ground-truth 3D gazing direction vector $\boldsymbol{n}_{\text{true}}$ is defined by a 3D vector from the eyeball center p_o computed by Eq. (10) to each 3D marker position. Then, the angular error of the 3D gaze estimation in each selected video frame is computed for each eye by

$$\theta = \arccos\left(\frac{\boldsymbol{n}_{\text{eye}} \cdot \boldsymbol{n}_{\text{true}}}{|\boldsymbol{n}_{\text{eye}}||\boldsymbol{n}_{\text{true}}|}\right) \tag{12}$$

On the other hand, a calibrated Tobii X120 Eye Tracker continuously outputs the estimated gazing position on the wall. Since Tobii is not designed to track the gaze of the two eyes separately, we compute the estimated 3D gazing direction vector $\boldsymbol{n}_{\text{eye}}$ as one 3D vector from the middle position of the two eye ball centers to each estimated marker position. Then the angular error of the Tobii's gaze estimation in each frame can be computed by Eq. (12).

Fig. 17 Template for iris detection.

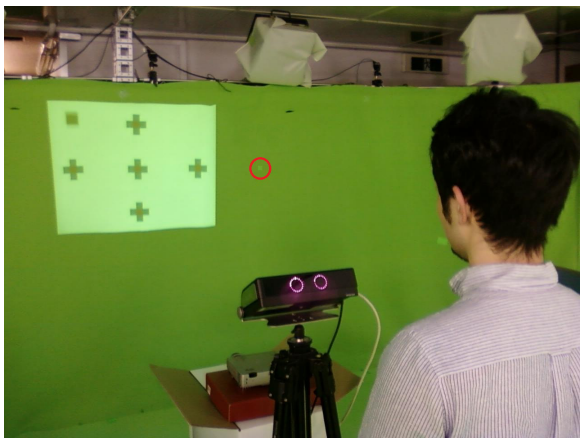


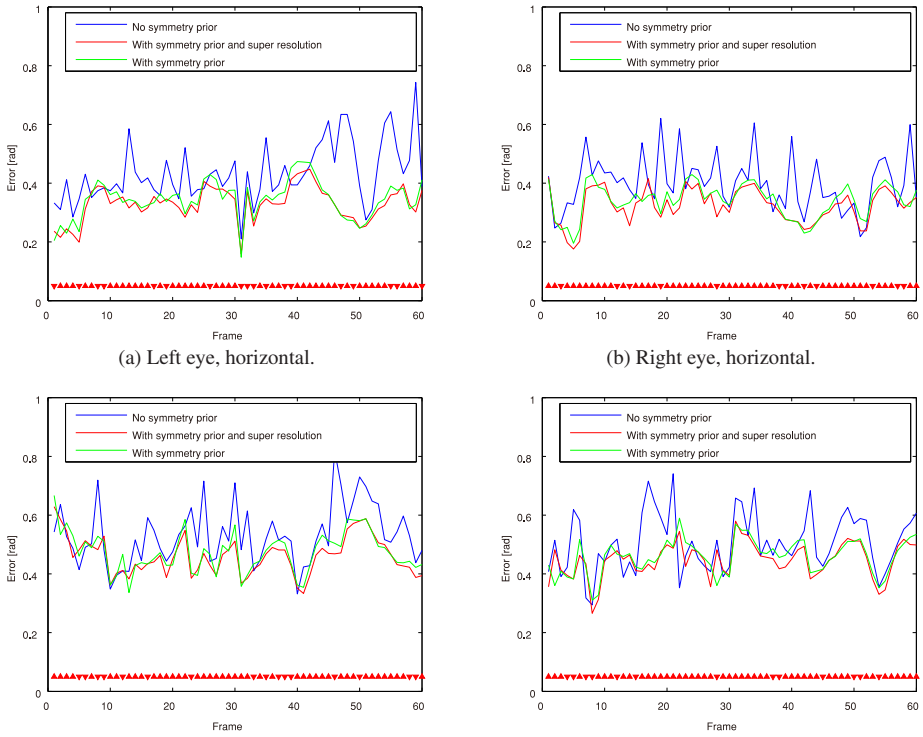
Fig. 18 Experiment with Tobii X120 Eye Tracker.

6.2.3 Experiment Results

In the analysis of the proposed method, the angular gaze estimation errors are evaluated for the left and right eyes as well as for the horizontal and vertical directions, respectively, which gives four different error evaluation results as shown in **Fig. 19** (a), 19 (b), 19 (c) and 19 (d). In each figure, three computational methods are compared: without the symmetry prior, with the symmetry prior alone, and with both the symmetry prior and the super-resolution image rendering technique. The horizontal axis in each figure denotes the selected frame IDs where the subject was stably looking at each marker. The upward and downward triangles at the bottom in each figure denote the signs (i.e., positive or negative) of the errors by the method with both the symmetry prior and the super-resolution image rendering technique. **Table 3** shows the average errors for the first and the third methods. **Table 4** compares the numbers of iris detection failures in 100 continuously captured frames. In all results, the symmetry prior improved the stability of the iris detection and the accuracy of the gazing direction estimation, while the improvement by the super resolution is limited. This is because the performance of the iris localization is not so accurate. As is well known, errors in the horizontal direction are much smaller than those in the vertical direction, because of the shape and movable range of human eyes. These results demonstrate the effectiveness and robustness of the presented method, while the accuracy of the gaze estimation is still limited. It should be noted that with the proposed method, we can perform gaze estimation on freely moving object as well as statically standing object. **Figure 20** is an example of using our gaze estimation method on Data A (Section 6.1.1), a

Table 2 Tobii X120 technical specifications.

Data rate	60 HZ
Latency	30–35 ms
Time to tracking recovery	Average 100 ms
Max gaze angles	35 degrees
Freedom of Head Movement	44 ∗ 22 ∗ 30 cm at 70 cm (Width ∗ Height ∗ Depth)
Tracker field of view	36 ∗ 22 ∗ 30 cm at 70 cm (Width ∗ Height ∗ Depth)
Top head-motion speed	35 cm/second
Eye tracking technique	Both bright and dark pupil tracking



(a) Left eye, horizontal. (b) Right eye, horizontal.

(c) Left eye, vertical. (d) Right eye, vertical.

Fig. 19 Gaze estimation errors of the proposed method. The upward and downward triangles at the bottom in each figure denote the signs (i.e., positive or negative) of the errors by the method with both the symmetry prior and the super-resolution image rendering technique.

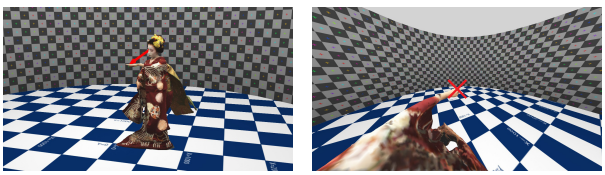
Table 3 Average gaze estimation errors of the proposed method.

	Left, horizontal	Right, horizontal	Left, vertical	Right, vertical
Original	0.4326	0.4002	0.5321	0.5063
Proposed	0.3297	0.3234	0.4610	0.4472

Table 4 Gaze estimation failures in 100 frames.

	Gaze estimation failure	Shape reconstruction failure	Iris detection failure
Original	5	0	5
Proposed	1	0	1

(a) Input multi-view images



(b) Estimated gazing direction in objective view (c) Estimated gazing direction in subjective view

Fig. 20 Gaze estimation on freely moving object. (a) input multi-view images, (b) estimated gazing direction in objective view, (c) estimated gazing direction in subjective view.

MAIKO performing traditional Japanese dance. The red arrow in Fig. 20 (b) and the red cross in Fig. 20 (c) illustrate the estimated gazing direction of the object. Since the full 3D shape of the object is reconstructed (Fig. 1 I), with the proposed method we can realize a subjective/first-person-view visualization, as is shown in Fig. 20 (c).

As for the gaze estimation results of Tobii X120 Eye Tracker, **Figs. 21** and **22** illustrate the angular errors of Tobii when working with markers inside and outside its effective region, respectively. The horizontal axis in Fig. 21 denotes the frame IDs where the subject is gazing at the five markers projected onto the wall, while the one in Fig. 22 denotes the frame IDs where the subject is gazing at the two markers manually set outside Tobii's effective region, as is described in Section 6.2.1. **Table 5** shows the average errors for Tobii X120 Eye Tracker. These results illustrate that Tobii performed the gaze estimation task with high accuracy when the markers are inside its effective region, while its accuracy dropped drastically when working outside its effective region. In addition, it should be noted that when the subject moved his head outside the freedom of head movement region, as is described in Table 2, or rotated the head over about 40 degrees, the Tobii X120 Eye Tracker failed to track the subject's eyes and gave no results, while the proposed method still succeeded to perform the frontal image synthesis and gaze estimation process.

Taking into account all these experimental results we could conclude that the proposed method performs no better than conventional method under the situation that the object's head move-

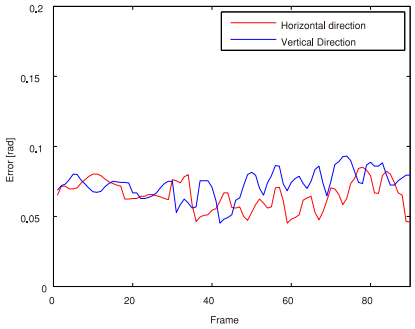


Fig. 21 Gaze estimation errors of Tobii X120 Eye Tracker inside its effective region.

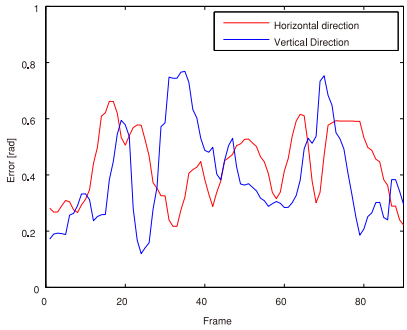


Fig. 22 Gaze estimation errors of Tobii X120 Eye Tracker outside its effective region.

Table 5 Average gaze estimation errors of Tobii X120 Eye Tracker.

	Horizontal	Vertical
Inside effective region	0.0651	0.0729
Outside effective region	0.4356	0.3996

ment is strictly limited. However, our work showed its advantage in the capability of dealing with freely moving object, which makes it possible to estimate the gazing behavior of humans in natural and complicated activities.

7. Conclusion

In this paper we proposed a novel 3D non-constrained and non-contact gaze estimation method that makes full use of the multi-view video data. The algorithm for the 3D gaze estimation consists of the 3D face area detection, the symmetry plane estimation, the accurate face surface reconstruction with the symmetry prior, the super-resolution frontal face image generation and the 3D gaze estimation based on the eyeball model. The algorithm worked stably to generate higher resolution frontal face images. The accuracy of the last process to estimate the iris position and gaze direction was also improved, while the absolute estimation accuracy was still limited. By comparing with a commercial gaze tracking device developed with conventional techniques, we have shown that our method exceeded others in the capability of performing robust gaze estimation on freely moving object, although its accuracy still needs to be improved.

For further studies, we should improve the gaze estimation algorithm by exploiting the temporal information. Also, introducing depth cameras into our system would make it possible to perform the proposed processing in real world environment with more complicated backgrounds and occlusions.

Acknowledgments The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported in part by the JSPS KAKENHI (23700204) and the JST-CREST project "Creation of Human-Harmonized Information Technology for Convivial Society."

References

[1] Morimoto, C.H. and Mimica, M.R.M.: Eye gaze tracking techniques for interactive applications, *CVIU*, Vol.98, No.1, pp.4–24 (2005).
[2] Weigle, C. and Banks, D.C.: Analysis of eye-tracking experiments performed on a Tobii T60, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol.6809 (2008).
[3] Laurentini, A.: How far 3D shapes can be understood from 2D silhouettes, *TPAMI*, Vol.17, No.2, pp.188–195 (1995).
[4] Franco, J.-S. and Boyer, E.: Exact polyhedral visual hulls, *Proc. BMVC*, Vol.1, pp.329–338 (2003).
[5] Kutulakos, K.N. and Seitz, S.M.: A theory of shape by space carving, *Proc. ICCV*, pp.307–314 (1999).
[6] Vogiatzis, G., Torr, P., Seitz, S. and Cipolla, R.: Reconstructing Relief Surfaces, *Proc. BMVC*, pp.117–126 (2004).
[7] Goldluecke, B. and Cremers, D.: Superresolution Texture Maps for Multiview Reconstruction, *Proc. ICCV*, pp.1–8 (2009).
[8] Tung, T., Nobuhara, S. and Matsuyama, T.: Simultaneous super-resolution and 3D video using graph-cuts, *Proc. CVPR*, pp.1–8 (2008).
[9] Yamazoe, H., Utsumi, A., Yonezawa, T. and Abe, S.: Remote gaze estimation with a single camera based on facial-feature tracking without special calibration actions, *Proc. 2008 Symposium on Eye Tracking Research and Applications* (2008).
[10] Cootes, T., Edwards, G. and Taylor, C.: Active appearance models, *TPAMI*, Vol.23, No.6, pp.681–685 (2001).
[11] Ishikawa, T., Baker, S., Matthews, I. and Kanade, T.: Passive driver gaze tracking with active appearance models, *Proc. 11th World Congress on Intelligent Transportation Systems*, pp.1–8 (2004).
[12] Guestrin, E. and Eizenman, M.: General theory of remote gaze estimation using the pupil center and corneal reflections, *IEEE Trans. Biomedical Engineering*, Vol.53, No.6, pp.1124–1133 (2006).
[13] Matsumoto, Y. and Zelinsky, A.: An algorithm for realtime stereo vision implementation of head pose and gaze direction measurement, *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, pp.1–8 (2000).
[14] Chen, J. and Ji, Q.: Probabilistic gaze estimation without active personal calibration, *Proc. CVPR*, pp.609–616 (2011).
[15] Viola, P.A. and Jones, M.J.: Robust Real-Time Face Detection, *IJCV*, Vol.57, No.2, pp.137–154 (2004).
[16] Fischler, M.A. and Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM*, Vol.24, No.6, pp.381–395 (1981).
[17] Nobuhara, S., Kimura, Y. and Matsuyama, T.: Object-Oriented Color Calibration of Multi-viewpoint Cameras in Sparse and Convergent Arrangement, *IPSJ Trans. Computer Vision and Applications*, Vol.2, pp.132–144 (2010).
[18] Furukawa, Y. and Ponce, J.: Accurate, dense, and robust multi-view stereopsis, *Proc. CVPR*, pp.1–8 (2007).
[19] Vogiatzis, G., Torr, P. and Cipolla, R.: Multi-view stereo via volumetric graph-cuts, *Proc. CVPR*, pp.391–398 (2005).
[20] Nobuhara, S. and Matsuyama, T.: Deformable Mesh Model for Complex Multi-Object 3D Motion Estimation from Multi-Viewpoint Video, *Proc. 3DPVT*, pp.264–271 (2006).
[21] Felzenszwalb, P. and Huttenlocher, D.: Efficient belief propagation for early vision, *IJCV*, Vol.70, pp.41–54 (2006).
[22] Kawaguchi, T., Rizon, M. and Hidaka, D.: Detection of eyes from human faces by Hough transform and separability filter, *Electronics and Communications in Japan*, Vol.88, No.5, pp.29–39 (2005).

Qun Shi received his B.Sc. in Computer Science from Northeastern University, China, in 2008 and M.Sc. in Informatics from Kyoto University, Japan, in 2011. Since then, he has been a Ph.D. student at Kyoto University. He has been engaged in Computer Vision.

Shohei Nobuhara received his B.Sc. in Engineering, M.Sc. and Ph.D. in Informatics from Kyoto University, Japan, in 2000, 2002, and 2005 respectively. From 2005 to 2007, he was a postdoctoral researcher at Kyoto University. Since 2010, he has been a senior lecture at Kyoto University. His research interests include computer vision and 3D video. He is a member of IPSJ, IEICE, and IEEE.

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, and human-computer interaction. 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 the Pattern Recognition Journal. He was awarded Fellowships from IAPR, IPSJ, and IEICE.

(Communicated by *Yoshio Iwai*)