Dynamically Visual Learning for People Identification with Sparsely Distributed Cameras

Hidenori Tanaka[1, 2], Itaru Kitahara[1], Hideo Saito[2], Hiroshi Murase[3], Kiyoshi Kogure[1] and Norihiro Hagita[1]

[1]Intelligent Robotics and Communication Laboratories, ATR,
2-2-2 Hikaridai, Keihanna Science City, Kyoto, Japan
{hidenori, kitahara, kogure, hagita}@atr.jp
[2]Graduate School of Science and Technology, Keio University,
3-14-1 Hiyoshi, Kouhoku-ku, Yokohama, Japan
saito@ozawa.ics.keio.ac.jp
[3] Graduate School of Information Science, Nagoya University,
Furo-cho, Chikusa-ku, Nagoya, Japan
murase@is.nagoya-u.ac.jp

Abstract. We propose a dynamic visual learning method that aims to identify people by using sparsely distributed multiple surveillance cameras. In the proposed method, virtual viewpoint images are synthesized by interpolating the sparsely distributed images with a simple 3D shape model of the human head, so that virtual densely distributed multiple images can be obtained. The multiple images generate an initial eigenspace in the initial learning step. In the following additional learning step, other distributed cameras capture additional images that update the eigenspace to improve the recognition performance. The discernment capability for personal identification of the proposed method is demonstrated experimentally.

1 Introduction

The recent deterioration of public safety is causing serious concern. Biometrics is one of the most promising technologies for alleviating this anxiety [1][2]. We are currently researching a form of biometrics that uses surveillance cameras set up in an actual space like a hospital or a railway station. For instance, we assume the hospital shown in Fig. 1. It is hoped that we obtain more appearance information at the entrance because at that point a suspicious person's invasion is obstructed.
Generally, because there is a broad field of view at the entrance, the images from different directions can be captured by using multiple cameras. If the monitoring system confirms that enough learning of an object's appearance has been performed, the automatic door opens and entry to the hospital is permitted. While the object is walking along the passage from the entrance to the sickroom, new images are captured with a surveillance camera arranged at each chosen position. The appearance information on the object is then updated by using the new images. When the object tries to enter the sickroom, another image of the object is captured by the surveillance camera set up in front of the sickroom. The personal identification processing is then

performed by using captured images, and when the result corresponds with the sickroom's authorization list, the automatic door opens and entry to the sickroom is permitted. A person's action history is generated with the processing of additional learning and identification. It is considered that different lighting conditions at each location has a strong influence on the accuracy of individual identification, though we assume to be able to control the lighting conditions almost constantly in indoor environments such as hospitals.

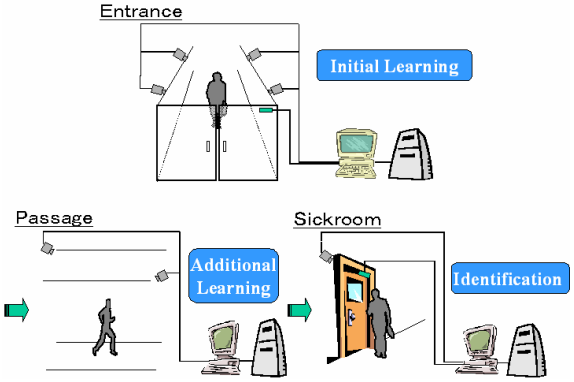


Fig. 1. Surveillance cameras in hospital

In this paper, we show a proposed method for dynamic visual learning based on a parametric eigenspace using sparsely distributed multiple cameras. We also describe some experiments to demonstrate the effectiveness of the proposed method.

2 Related Works

As people generally utilize facial appearances to identify individuals, the human face has potential for use as the most important type of information for biometric technology, making face recognition is one of the most important reasons for installing surveillance video sensors [3][4]. Most of these sensors demand a frontal or near-frontal facial view as the input appearance, and extract points of interest for the identification process (e.g., eyes, brows, nose and mouth). However, it is not always possible to capture the desired appearance with practical surveillance cameras. Therefore, in order to achieve a high recognition rate, the systems have to severely restrict people's activities, as in a portrait studio.

Parametric eigenspace is a well-known method for identifying objects with various appearances from images [5]. In order to generate a parametric eigenspace that achieves accurate identification, a number of different appearances, which can be collected by densely distributed multiple cameras, are generally required. However, it is not practical to set up a dense network of surveillance cameras around objects in the real world; general-purpose surveillance cameras are sparsely distributed, because the primary objective of the cameras is to monitor the widest area possible.

The objective of this paper is to realize a dynamic visual learning method based on parametric eigenspace to identify people captured with sparsely distributed multiple surveillance cameras. If we simply generate an eigenspace with a small number of

sparsely distributed images, it is not possible to identify people from various viewing angles because the eigenspace cannot effectively learn the variety of appearances. Murase et al. attempted to fill the gap between the multiple images with a spline interpolation technique in a roughly generated eigenspace [6]. In our case, however, the gap is much larger than the one they dealt with in their research. The reason why spline interpolation does not work well with sparsely distributed images is that changes in the captured object's appearance are not considered in the interpolation process. The Virtualized Reality popularized by Kanade synthesizes a novel view with a 3D shape reconstructed from images captured with multiple cameras [7]. This technique makes it possible to interpolate multiple images by considering changes in appearance. In our proposed method, we mainly employ this technique to virtually capture multiple images and to generate an initial eigenspace. However, we need to modify this technique by simply using a 3D face model provided by the Galatea project [8], rather than recovering the 3D shape of the object from the captured multiple images, because it is still difficult to recover the 3D shape of the object from sparsely distributed multiple surveillance cameras.

3 Proposed Method for People Identification with Sparse Multiple Cameras

As illustrated in Fig. 2, the proposed method consists of two phases. We call the first phase the "initial learning phase," and the second one the "additional learning phase."

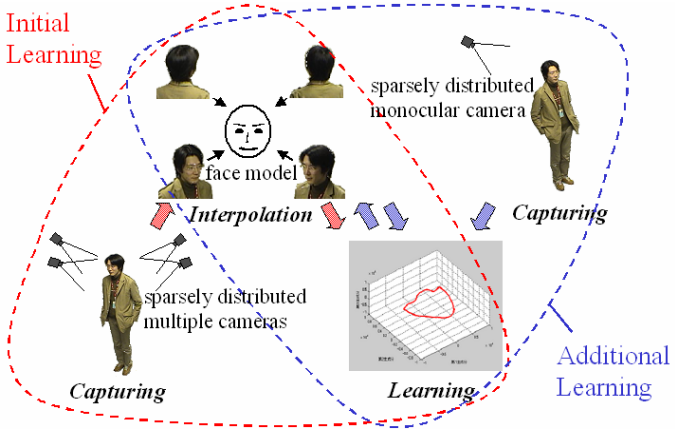


Fig. 2. Parametric eigenspace method with sparsely distributed multiple cameras

Initial Learning: In this phase, a view interpolation technique is used to generate an initial eigenspace. The surrounding sparsely distributed multiple cameras capture the target object at the same time. The 3D visual hull of the object is then reconstructed by merging the multiple images using the shape from a silhouette method [9]. A simple 3D face model is fitted to the visual hull to mask the face region, and as a result, a 3D shape model is estimated. This method virtually captures multiple images by interpolating the appearance information of sparsely distributed images with the 3D shape model, and generates an initial eigenspace.

Additional Learning: To improve the method's ability to identify individuals in the eigenspace, the additional learning phase dynamically updates the eigenspace generated in the first phase. The extrinsic parameter of the additional capturing cameras is estimated as a relative pose with respect to the object. Then, the captured image is projected onto the 3D shape model with the parameter as texture information to improve the appearance of the interpolated images. By regenerating the eigenspace of the updated image data set, the discernment capability for personal identification of the eigenspace is improved.

4 Initial Learning Phase

4.1 Extraction of Head Region

As illustrated in Fig. 3, we set up a camera that looks down from the ceiling of the target space. Koyama et al. estimated 3D positions *(X, Y, Z)* from the 2D coordinates in an overhead image *(u,v)* while assuming that all target objects are at a constant height *Y* [10]. Under this assumption, a homographic transformation *H* is calculated that projects 2D coordinates in the overhead image onto a 3D plane. Eq. (1) is the equation of the projection. However, this assumption imposes a limitation on detecting the objects.

$$\lambda[X \quad Z \quad 1]^{T} = \mathbf{H}[u \quad v \quad 1]^{T} \tag{1}$$

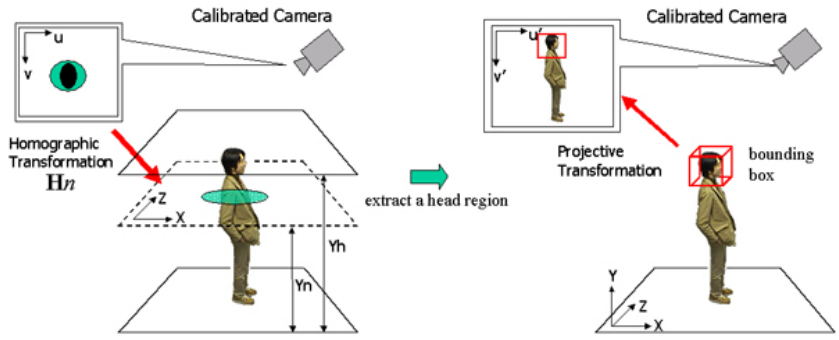


Fig. 3. Extraction of the head region

We improve this plane-based object-tracking method to detect the object's 3D position with arbitrary height information [11]. In this method, two base-planes are placed in 3D space, one on the ground *(height=0)* and the other at height *Yh*, after which the homographic transformations *H0* and *H1* are calculated. If the height of a new plane is given as *Yn*, the homographic transformation *Hn* is estimated by interpolating *H0* and *H1*, as in Eq. (2).

$$\mathbf{H}_{n} = ((Y_{h-}Y_{n})\mathbf{H}_{0} - (Y_{n})\mathbf{H}_{1}) / Y_{h} \tag{2}$$

The segmented foreground region is projected onto the 3D plane-*n* by changing the height *Yn* from *0* to *Yh*. If the target object stands vertically like a human, the projected foreground region always includes a certain point *(X,Z)* on plane-n where the actual 3D object exists. By merging all of the n-planes (e.g., by an AND operation), the 3D position of the target object is estimated.

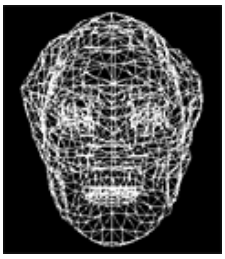


Fig. 4. A simple 3D face model

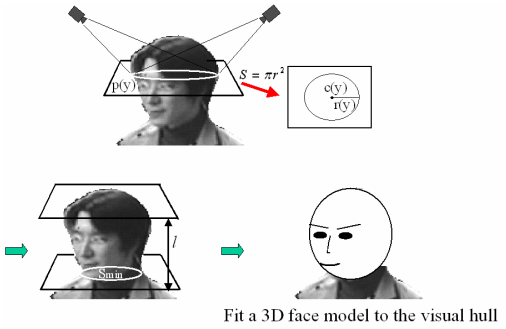


Fig. 5. 3D model estimation

4.2 3D Shape Model Estimation and View Interpolation

The accuracy of the estimated 3D shape seriously affects the quality of the interpolating images. To solve this problem, we employ a simple 3D face model provided by the Galatea project and fit the model to the visual hull to mask the face region. Fig. 4 shows the wire frame model of the 3D face model.

As illustrated in Fig. 5, we set a horizontal plane *p(y)* and project all of the foreground regions in all of the multiple images that have been calibrated in advance. The sliced 3D shape of the object is estimated as the overlapped region of all the projected regions [12]. We calculate the size and position of the captured head as the radius *r(y)* and the center of the circle *c(y)=(X,Y,Z)* by fitting a circle to the estimated shape on the plane, and execute the same process while shifting the horizontal plane along with the vertical axis to cover the head region. The head height *l* is estimated by searching for the minimum nonzero radius and the highest position of the head. As the right-hand side of Fig. 5 shows, we scale the 3D face model up/down with respect to head height and put the 3D model at the center of the head region. With this scaling process, we can reflect individual differences in head size.

The input multiple images are texture-mapped onto the estimated 3D face model using Projective Texture Mapping [13]. We render interpolation images while rotating a virtual camera around the 3D model in 1° increments. The blending parameter of each image (texture) is then calculated using the distance from the input multiple cameras to the viewpoint currently being interpolated.

4.3 Parametric Eigenspace Generation

4.3.1 Normalization

Normalization consists of two steps: scale normalization and brightness normalization. In scale normalization, the extracted head region is resized to a square region

(e.g., 128×128 pixels) with its center at the top of the head. In brightness normalization (Eq. (3)), each image $\hat{x}_i$ is transformed to a normalized image x_i. In this normalizing process, our method has an advantage in that it is possible to completely control the conditions while generating the input image set, because they are synthesized images.

$$\mathbf{x}_i = \hat{\mathbf{x}}_i / \left\| \mathbf{x}_i \right\| \tag{3}$$

4.3.2 Creating the Initial Eigenspace

To compute the eigenspace, we first subtract the average *c* of all images in the image set from each image in the set as shown in Eq. (4), where *N* is the total number of images in the image set. Then, to compute the eigenvectors of the image set, we define the covariance matrix *Q* also given in Eq. (4). The eigenvectors e_i and the corresponding eigenvalues λ_i of *Q* are to be computed by solving the eigenvector decomposition problem using Eq. (5). All of the eigenvectors of *Q* constitute a complete eigenspace. However, only a small number of eigenvectors is generally sufficient for capturing the primary appearance characteristics of objects. These k eigenvectors correspond to the largest k eigenvalues of *Q* and constitute the eigenspace. The number k of eigenvectors to be computed is selected based on recognition ability.

$$\mathbf{Q} = \mathbf{X}\mathbf{X}^{\mathrm{T}}, \quad \mathbf{X} = \left[\mathbf{x}_1 - \mathbf{c}, \mathbf{x}_2 - \mathbf{c}, \cdots, \mathbf{x}_N - \mathbf{c} \right] \tag{4}$$

$$\lambda_i \mathbf{e}_i = \mathbf{Q}\mathbf{e}_i \tag{5}$$

Each image x_i is projected onto the eigenspace. This is done by subtracting the average image *c* from x_i, and then finding the dot product of the result with each of the *k* eigenvectors, or dimensions, of the eigenspace as in Eq. (6), where *i* indicates the pose parameter. The result is a single point in the eigenspace, and by projecting all of the image sets, we obtain a set of discrete points. Pose variation between any two consecutive images is small; as a result, their projections in eigenspace are close to one another. Such a sampling creates a continuous loop in the eigenspace.

$$g_i = \left[\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_k \right]^{\mathrm{T}} \left(\mathbf{x}_i - \mathbf{c} \right) \tag{6}$$

5 Additional Learning Phase

5.1 Global Search

A surveillance camera captures a person who has already registered his/her appearance information in the initial learning phase. We extract the normalized head region from a captured image y_j with the above-described method, and project the region onto the calculated eigenspace. Concretely, the average *c* of the entire image set used to compute the eigenspace is subtracted from the input image y_j. The resulting image is projected onto eigenspace to obtain a point z_j. The equation for this the projection is Eq. (7). In order to search for the interpolated image most similar to the input im-

age, we calculate the Euclidean distance of each eigenvector in the eigenspace between the input image z_j and the view-interpolated images g_i. The parameter of the interpolated image that has the most similar eigenvector to the input image's vector is estimated as the rough relative observing orientation.

$$z_j = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_k]^{\mathrm{T}} (\mathbf{y}_j - \mathbf{c}) \quad (7)$$

5.2 Local Search

Since the activity of the captured people is not controlled, their poses might be different from the poses in the initial learning phase. Thus, the estimated observing orientation contains a measure of error. In this section we describe a method for correcting the estimation error.

5.2.1 Generating a Synthetic View

Once a 3D model has been generated, it is possible to render synthetic views of the modeled face with various rotation and translation parameters of the cameras. We assume the 3D model to be fixed, and the camera moves relative to it. In order to render images that have a slight difference in appearance with the matched image in the global search, synthetic view generation is repeated for a range of rotations around the *x*, *y* and *z* axes (the *x* axis is through the neck, the *y* axis is through the head, and the *z* axis is perpendicular to the *x* and *y* axes). Typically we use plus or minus 30°, plus or minus 5°, and plus or minus 10° around the *x*, *y* and *z* axis, respectively, quantized in 5° intervals, for a total of 195 synthetic views. Fig. 6 shows some sample generated images.

Fig. 6. Synthetic views that have slight differences in appearance from the matched image in a global search.

5.2.2 Matching Against Synthetic Views

To find the best match, we compare the input image with each of the synthetic views. Consider the input image *I* that is being matched against a synthetic image *S*. The goodness of the match *M* between the two is found by computing Eq. (8), where *I(I,j), S(i,j)* are the image intensity at pixel *(i,j)* in the input and synthetic images, respectively. This is the well-known SAD method. The best-matching synthetic view is the one that minimizes this score.

$$M(s,t) = \sum | I(i,j) - S(i+s, j+t) | \quad (8)$$

We are not, however, aiming to obtain the best-matched image but to get the camera parameters. To do this we use a downhill simplex method [14], which is a

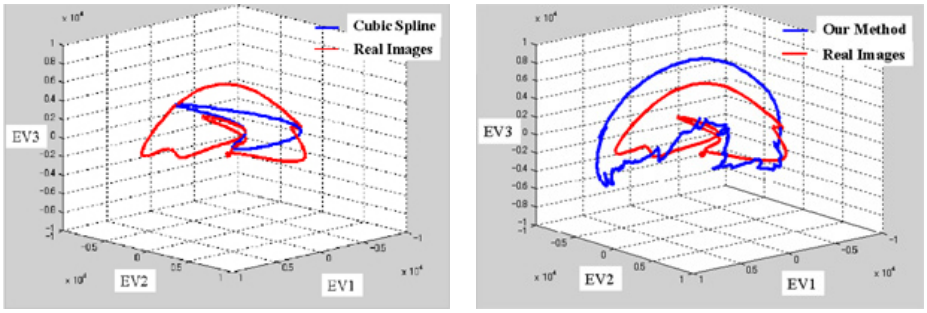
multi-dimensional optimization method, to estimate the camera parameters in order to minimize Eq. (8). To avoid a local minimum solution, we start from random values, and then pick the solution corresponding to the lowest minimum value.

5.3 Updating the Eigenspace

By projecting the input image with the estimated camera parameters, the texture information of the captured object's 3D shape model is updated. Then, as in the processing in Section 4.2, the interpolating images are regenerated while rotating a virtual camera around the 3D model. We thus again calculate an eigenspace with the updated image data set. If the appearance information of the 3D model becomes more accurate with additional texture mapping, the discernment capability for personal identification of the eigenspace improves further. We demonstrate the effectiveness of the proposed method in the next section.

6 Experiments

In these experiments, all images are captured by Sony network cameras (SNC-Z20) with a 640x480-pixel image size. All eigenspace figures in this section show only three of the most significant dimensions of the eigenspace since it is difficult to display and visualize higher-dimensional spaces. In other processes, the number of eigenspace dimensions used is 25, and in this case the accumulation-contributing ratio exceeds 95%.



(a) Interpolation using the cubic spline function (b) Interpolation using our method

Fig. 7. Comparison between the two types of interpolation methods

6.1 Interpolation Method Results

To evaluate the effect of the viewpoint interpolation in the proposed method, we compared the locus of the proposed method with the locus interpolated using a cubic spline function. In this experiment, the real images captured at intervals of about 22.5 ^{0} were assumed to be the standard. Fig. 7(a) illustrates the result of the real images that are regarded as the standard and the result of interpolation using the cubic spline function. In this figure, to improve the visibility of the loop features, we have interpolated the real images with the cubic spline function that is generally used to interpolate eigenspace. Fig. 7(b) represents the result of the real images that are regarded as the standard and the result of interpolation using a 3D model. Comparing Fig. 7(a)

with Fig. 7(b), we see that interpolation using a 3D model is more complex than that using the cubic spline function for the real images. From this result, when we have only sparsely input images, it can be said that interpolation using a 3D model can create a higher discernment capability eigenspace than interpolation using the cubic spline function.

6.2 Additional Learning Results

Next, we examine how the additional learning process updates the initial interpolation images. The upper row in Fig. 8 shows interpolated images from the four initially captured images. With our proposed additional learning method, the encircled regions are updated by newly captured images. On the bottom row of Fig. 8 we can see that the realism of the interpolated images is improved by the replacements provided by additional learning.

Fig. 8. Result of additional learning phases

6.3 Results of Discernment Capability

We have already experienced how the discernment capability among four persons varies as replacement is performed. In this experiment, first, one person from a group of four is chosen as the identification object, and interpolation images of the person are subsequently generated with a 3D model. Then, 50 face images of all four people are projected to the eigenspace generated by using the interpolation images, and we calculate the distance in the eigenspace among the interpolation images and the projection point. The distance (threshold) is obtained by comparing the projection point of the 50 face images of the four persons with the projection point of another 50 face images of the same four persons. Fig. 9 shows how the discernment capability among the four persons varies as replacement is performed. We can see that the discernment rate improves as additional learning progresses, and that discernment capability has improved. However, the discernment rate decreased when the number of additional images became four from three. We think this loss of performance occurs due to the gap in the texture mapping and errors in extraction.

7 Conclusions

We proposed a learning method for parametric eigenspace using sparsely distributed multiple cameras. We have demonstrated that the discernment capability of the initial eigenspace is improved by repeating the updating process, and that interpolation using a 3D model more closely resembles the real image than interpolation using the cubic

spline function. Future work will include reducing errors in extraction and a method to put together various pieces of information for personal identification. This research was supported in part by the National Institute of Information and Communications Technology.

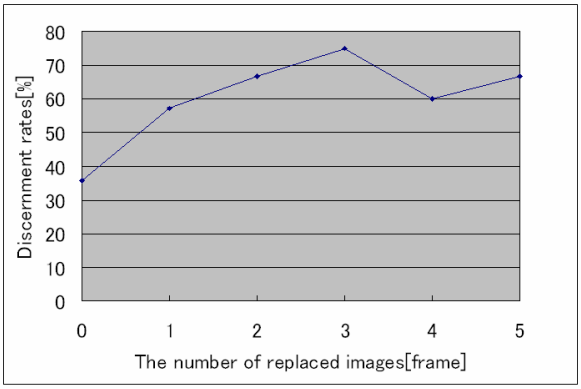


Fig. 9. Number of updates vs. Discernment capability results

References

[1] A.K. Jain, S. Pankanti, S. Prabhakar, L. Hong, A. Ross, Biometrics: A Grand Challenge, *Proc. of IC PR*, 2004, Vol. 2, pp. 935-942

[2] A. Pentland, Looking at People: Sensing for Ubiquitous and Wearable Computing, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2000, Vol. 22, No. 1, pp. 107-118.

[3] S. Lao, T. Kozuru, T. Okamoto, T. Yamashita, N. Tabata, M. Kawade, A fast 360-degree rotation invariant face detection system, *ICCV,* 2003

[4] http://www.identix.com/products/pro_security_bnp_argus.html

[5] H. Murase, S.K. Nayar, Parametric Eigenspace Representation for Visual Learning and Recognition, *Workshop on Geometric Method in Computer Vision, SPIE*, 1993, pp. 378-391.

[6] H. Murase, S.K. Nayar, Illumination Planning for Object Recognition Using Parametric Eigenspaces, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1995, Vol. 16, No. 12, pp. 1219-1227.

[7] T. Kanade, P.J. Narayanan, and P.W. Rander, Virtualized reality: concepts and early results, *Proc. of IEEE Workshop on Representation of Visual Scenes*, 1995, pp. 69-76.

[8] http://hil.t.u-tokyo.ac.jp/~galatea/

[9] A. Laurentini, The Visual Hull Concept for Silhouette-Based Image Understanding, *IEEE Trans. on Pattern Analysis and Machines Intelligence,* 1994, Vol. 16, No. 2, pp. 150-162.

[10] T. Koyama, I. Kitahara, Y. Ohta, Live Mixed-Reality 3D Video in Soccer Stadium, *Proc. of ISMAR,* 2003, pp. 178-187

[11] I. Kitahara, K. Kogure, N. Hagita, Stealth Vision for Protecting Privacy, *Proc. of ICPR*, 2004, Vol. 4, pp. 404-407

[12] I. Kitahara, H. Saito, S. Akimichi, T. Ono, Y. Ohta, and T. Kanade, Large-scale Virtualized Reality, *CVPR, Technical Sketches*, 2001.

[13] C. Everitt, Projective Texture Mapping, *White paper, NVidia Corporation*, 2001.

[14] J.A. Nelder and R. Mead, A Simplex Method for Function Minimization, *Computer Journal,* 1965, Vol. 7, pp. 308-313.