Offset-Block Matching of Multi-view Images for Ray-Space Interpolation

Mehrdad Panahpour Tehrani †, Toshiaki Fujii (member)‡, Masayuki Tanimoto (member)‡

Abstract  In this research, we address the problem of finding the best correspondence in matching algorithms in the edge area and we propose offset-block matching for ray-space interpolation. Conventional block matching methods are pixel-based, block-based and hierarchical methods. The pixel-based matching method performs well and satisfactorily detects correspondence in the edge area, when the search area is small. Block-based matching methods detect better than pixel-based methods where is a large disparity, however the detecting of correspondence in the edge area is not as accurate. The hierarchical block matching method performs relatively well in all areas. The proposed offset-block matching however combines pixel-based and block-based matching methods by limiting the search area and obtains better results than conventional matching methods. The proposed method has simultaneous advantage of both the pixel-based and the block-based matching methods. Furthermore, this method is suitable for distributed architecture of a video rendering system like FTV, when we need real-time interpolation.

Key words: Offset-block matching, ray-space interpolation, epipolar plane image (EPI), pixel-based and block-based matching.

1. Introduction

In this paper, we propose a new block matching method for ray-space interpolation. We have proposed Free viewpoint TeleVision (FTV) system1. In this system user can freely control the viewpoint position of any dynamic real-world scene using ray-space234 interpolation. Ray-space interpolation is a technique that generates the missing ray of a scene, by corresponding search. Ray-space interpolation requires computation of correspondence among points in stereo images in order to generate the arbitrary view. In FTV system distributed processing4 scheme is optimal for corresponding search problem in ray-space interpolation. Hence, a matching method for ray-space interpolation, suitable for distributed architecture of FTV is also needed.

Matching method is used in many areas of image processing, such as multimedia, and vision fields. In this paper our focus is on the application of block matching to the computation of correspondence under epipolar plane image (EPI) constraint. Correspondence search methods are based on finding the best match in the maximum disparity or search area, with the least error, usually by minimizing either mean square error (MSE), or mean absolute error, which is easier to compute. In the case of ray-space interpolation, the captured images are distorted and also need to be rectified5 before interpolation. In order to find the best match in two dimensions is computationally expensive. Removing the lens distortion and rectifying of the raw images are not the aim of this research, however the search problem will reduce to a one-dimensional search along raster line of the images with one pixel accuracy in vertical direction. Considering the benefits of the above preprocessing, the matching algorithms in this work are restricted for multi-view images after the lens distortion removal and rectification, under the EPI constraint.

Many correlation-based techniques including pixel-based, block-based, and hierarchical methods try to solve the problem of finding the best correspondence during the matching process. The pixel-based method is simply based on search for the best matched pixels in the stereo image pairs using the absolute difference. This matching method has a good detection of the best-matched pixels, especially in the edge area when the search area is small. Therefore, in the large disparity cases or when the search area is large, the performance
of the pixel-based matching method for ray-space interpolation is not good.

Several block-based matching methods\(^{6,7,12}\) have been proposed for different applications. A full search block matching method\(^8\) for ray-space interpolation under the EPI constraint has been proposed using MSE as correlation measure. This block matching method is similar to fixed size block matching (FSBM) method\(^8\) where the block is fixed and similar to variable size block matching (VSBM) methods\(^9,12\) where multiple block sizes are used in matching algorithm. Another well-known block matching method is hierarchical block matching method\(^13\). In general, block matching methods\(^{6,7,12,13}\) have always better correspondence detection than the pixel-based method except in the small disparities for ray-space interpolation. Using the block-based matching methods to check the correspondence in the edge area results miss-detection. The hierarchical block matching method has better performance compared to normal full search block matching methods, however, its performance is not optimal and needs to be improved.

To solve the miss-detection in the edge area using the block-based methods, smaller block sizes can be used. However it will cause miss-detection in the pixel-based method when the disparity is increased. Therefore, a matching method is needed that can find the best correspondence in the edge area, while having a good performance in large disparities. Such a matching scheme combines the advantages of both the pixel-based and the block-based methods.

In the case of large disparities the use of the block-based matching methods cannot be avoided. In the block-based matching methods the only main parameter is the block size. In order to capture the benefits of both the pixel-based and the block-based matching methods, we have defined an additional parameter, namely offset size. We propose offset-block matching method, a new matching scheme. In the proposed method, for a given block size, all blocks in the pre-assigned searching length or maximum disparity between stereo images are grouped. Also the blocks in each group are not overlapped. Moreover, the number of groups depends on the offset size between groups. In the matching algorithm, first, the best-matched block of each group is found using MSE and is called candidate block of the group. Then, the overlapped parts among all candidates are found and are called candidate pixels. As a result of grouping, the search area in the candidate pixels is small, thus, the pixel-based method can be used for the candidate pixels. Therefore, proposed offset-block matching method gains the advantages of the both methods, by adding the offset size as a parameter to the matching algorithm. Comparison of the performance of the proposed method and the conventional matching methods shows that the offset-block method outperforms the conventional schemes in all areas for all disparities.

The remaining of this paper is organized as follows. In section 2 ray-space methods are explored. In section 3, ray-space interpolation is explained. In the section 4 and 5 the conventional and the proposed matching methods are introduced, respectively. The experimental results are shown in section 6. The outperforming points of the proposed method are discussed in section 7. Section 8 concludes this paper.

## 2. Ray Space Representation

Ray-space was originally proposed as a common data format for 3D image communication\(^{2,3}\). A similar idea has been proposed in computer graphics field for generating photo-realistic images into a computer generated virtual world\(^{4,5,12}\). Another technique, called Lumigraph\(^{5,12}\), has been proposed in the computer graphics field and has been widely used for generating photo-realistic images. These techniques are used to produce computer generated virtual worlds. Both these techniques are based the idea that a view image from an arbitrary viewpoint can be generated from collection of real view images. Ray-space method describes 3D spatial information as the information of the a ray, which transmits in the space. **Fig. 1** shows an example of the definition of ray-space. Let \((x, y, z)\) be three space coordinates, and \((\theta, \phi)\) be the parameters of direction. Symbols \(x\) and \(y\) represent the point of the intersection of the ray with the XY-plane. Also \((\theta, \phi)\) denote the angles that the rays passing through the XY-plane make with the Z-axis in horizontal and vertical directions, respectively. These angles are illustrated in **Fig. 1(a)**. A ray going through free space is uniquely designated by the coordinates \((x, y)\) of the intersection with the plane \(z = 0\) and the angles \((\theta, \phi)\). These ray parameters construct a 4D space. In fact, this is a 4D subspace of a 5D ray-space \((x, y, z, \theta, \phi)\). In this ray-parameter, we define a function \(f\) whose value corresponds to the intensity of the specified ray. Thus, all the intensity data of rays can be expressed by equation (1). This ray parameter is called the ray-space.
In general, the rays that pass through a specific plane can be captured using a pinhole camera. When we locate a camera at \( (x_0, y_0, z_0) \), the intensity data of rays that go through the pinhole are given by equation (2).

For simplicity, we consider only the 2D subspace \( f(x, y) \) of the 5D ray-space, where the vertical parallax \( \theta \) and vertical position \( y \) are neglected. This 2D plane of the ray-space data is Epipolar Plane Image (EPI), which is the cut of ray-space data, parallel to \( xu \)-plane for a given \( y \). In other words, EPI is the intersection of the epipolar plane with the camera plane in stereovision, which is a well-known concept in computer vision field. If there exists an array of cameras sharing a common epipolar plane, then the intersection of the common epipolar plane with all the camera planes can be captured in a single image. This image is called EPI. Let \( (X, Z) \) be real-space coordinates, and \( x \) and \( u \) be the ray-space coordinates as shown in Fig. 1(b). The rays that pass through a point \( (X, Z) \) in the real-space form a line in the ray-space given by equation (3).

\[
f(x, y, \theta, \phi), \quad -\pi \leq \theta < \pi \quad \text{and} \quad -\frac{\pi}{2} \leq \phi < \frac{\pi}{2} \quad (1)
\]

\[
f(x, y, z, \theta, \phi), \quad x = x_0, \quad y = y_0, \quad z = z_0 \quad (2)
\]

This characteristic that the view image, as described above, corresponds to a cross section image of the ray-space data, is the most important characteristic of the 4D ray-space representation.

The view acquisition/display process is considered to be the recording/extracting process of the ray-space data along the locus. In the acquisition process, we sample the ray data in the real space and record them as the ray-space data along the locus as shown in Fig. 1(b). In the display process, we cut the ray space along the locus and extract the section image of the ray space as shown in Fig. 1(c). Fig. 1(d) shows the generated images of section 1 and section 2. However, in the case of not-dense ray-space data interpolation is needed. In such a case interpolation generates the missing rays between two EPI lines. The EPI lines are in \( x \) direction for a given \( u \) as 1D subspace of 5D ray-space data are the same as the intersection of epipolar plane with one camera plane in computer vision field. The interpolation task should be done on 2D ray-space data or EPI. The interpolation generates an EPI line between two EPI lines. In the next section, the ray-space interpolation is explained under the EPI constraint.

3. Interpolation

The ray-space interpolation technique we adopted here is based on the adaptive filtering interpolation\(^6\)\(^7\). This technique was designed to replace a simple linear interpolation technique, which is suitable for input images with camera axis parallel to the same plane. To make the camera axis parallel, lens distortion removal and rectification\(^5\) are needed. Multi-view images interpolation is performed after capturing, lens distortion removal and rectification processes for each image. In this way, the technique is applied without much difficulty.

The interpolation starts with the preparation of a set of filters corresponding to the best match or disparity. The input images are converted into 2D ray-space data or EPI after performing lens distortion and rectification. Each EPI is then up-sampled. Hence, we can apply the filter set to the input ray-space data as shown in Fig. 2 after finding the best matched direction for interpolation.

Note that the matching algorithm consists of searching for the best matched area for a given location of the \( \text{"pixel to be interpolated"} \) between two EPI lines. The searching length is equal to maximum disparity between two stereo images, which depends on the distance between the EPI lines.

\[
\begin{align*}
\text{EPI line 1} & \quad \leftrightarrow \quad \text{EPI line 2} \\
\text{(a) original EPI} & \quad \leftrightarrow \quad \text{(b) up-sampling} \\
\text{filter} & \quad \leftrightarrow \quad \text{(c) filtering} \\
\text{interpolaton} & \quad \leftrightarrow \quad \text{(d) interpolation}
\end{align*}
\]
between two cameras (i.e., camera interval) and the distance between object and camera plane.

For each "pixel to be interpolated", the matching algorithm aims at choosing the best direction (i.e., optimum filter) or disparity for interpolation. The direction that results in the lowest error is selected for interpolation. Then, the "pixel to be interpolated" value is calculated using the matched pixel values of the two EPI lines which pass the "pixel to be interpolated". The matched pixels are the "best corresponded pixel" pair for interpolation. The problem of finding the "best-corresponded pixel" is solved in the matching algorithm. The main advantage of this method is that the approximate geometric models are not required prior to the interpolation mechanism. Moreover, the scene complexity does not have any effect on the interpolation speed.

4. Conventional Matching Methods

4.1 Pixel-based Matching Method

If the captured multi-view images are parallel after lens distortion removal and rectification, the complexity of the search for correspondence is decreased to one dimension along the raster line of the two views or the two EPI lines. The search in this method is simply limited to search for the "best-corresponding-pixel" pair on the two EPI lines. The interpolation is performed in the assigned maximum disparity for a given location of the "pixel to be interpolated". Fig. 3 shows the upsampled EPI with two EPI lines. Each EPI line is a pixel row of a viewpoint. In the pixel-based matching, we have to find the 'best-corresponded pixel' pair in "EPI line1" and "EPI line2" with minimum absolute value difference. In Fig. 3, one step of checking the correspondence in the pixel-based method is depicted. For example pixel "a" belongs to viewpoint 1 (i.e., EPI line 1), and pixel "b" belongs to viewpoint 2 (i.e., EPI line 2). Hence, we need both location of "a" and "pixel to be interpolated" to compute the location of "b". The line passing through pixel "a", "pixel to be interpolated" and pixel "b" determines the direction for checking the correspondence. The absolute difference values of all the pairs of pixels (e.g., pixel "a" and pixel "b") in two EPI lines are considered as similarity error. After calculating all the similarity errors in the searching length, the pair of pixels with the lowest error is applied to linear interpolation. This matching method has only good performance in the edge areas where the searching length is small. Therefore, when the search area is increased, the matching algorithm detects the local minima, which causes miss-detection or degradation in interpolation quality.

4.2 Block-based Matching Method

As it has been explained in the previous section, the block matching can also be applied to the blocks which have one pixel in u direction as shown in Fig. 4 (e.g., block "A" and block "B").

The block-based matching method consists of searching for corresponding blocks in the two EPI lines to find the best-corresponding pixel pair for interpolation in the assigned maximum disparity. Fig. 4 shows an EPI with two EPI lines in which is up-sampled. In this figure, one step of checking the correspondence in the block-based method is depicted. For example, block "A" belongs to viewpoint 1 (i.e., EPI line 1), and block "B" belongs to viewpoint 2 (i.e., EPI line 2). Hence, having the locations of block "A" in EPI line1 and "the pixel to be interpolated", the location of block "B" in EPI line 2 can be found. The line passing through block "A", "pixel to be interpolated" and pixel "B" determines the direction for checking correspondence. The MSE values of all block pairs (e.g., block "A" and block "B") in two EPI lines are considered as similarity errors. After
calculating all similarity errors in the searching length, the middle pixel of the block pair with the lowest error is applied to the linear interpolation. If the number of pixels in blocks are even, the average value of the two middle pixels of each block is used for interpolation.

The block-based matching method performs well when the searching length or the camera interval is large. However, independent of the searching length, it has miss-detection in the edge area. In other words, the block-based method has a better performance in large search length than the pixel-based. However in the small search area the pixel-based method can perform better than the block-based matching method.

4.3 Hierarchical Block Matching Method

Same as block-based method, hierarchical block matching method finds the best-corresponding blocks in the two EPI lines. However, this method will not directly use the middle pixel of the block for interpolation. In the next step, the algorithm shortens the search length to the found block length. Then, by reducing the block size (e.g., one pixel), it finds the best-matching block in the first found block. The same procedure is continued until one pixel is chosen. In this step, the algorithm uses that pixel for interpolation. The hierarchical block matching has better performance than block-based method in the edge area. However the pixel-based method is still better in small search area. Furthermore, the pixel-based method has a relatively good detection in the edge area, because it does not use the middle pixel value of the block to interpolate. In other words, this method limits the search area to one block size. Hence, if a matching method can obtain a smaller searching area it will be able to perform even better than the hierarchical block matching method. The matching method proposed in this paper, described in the following section, reduces the searching length to less than one block size.

5. Proposed Matching Method

To avoid the miss-detection in edge area, we aim to gain the advantage of the block-based matching method, and limit the search area to be able to use pixel-based method, efficiently. Block size "B" and offset size "O" are the two parameters in the proposed matching algorithm. For a given "B" and "O" and maximum disparity, blocks of EPI line 1 are grouped, as shown in Fig. 5. A "group" is a collection of non-overlapped blocks, where there is not any gap between blocks. In the example of Fig. 5, "B=6", "O=1" and Maximum disparity is 19 pixels. The next step is to find the best-matched block in each group of EPI line 1 to the block in EPI line 2. We call the best-matched block, "candidate block" of each group. The candidate blocks have the least MSE in each group. Finding the candidates blocks is the same as the procedure mentioned in section 4.2 for the block-based matching method. In other words, having the location of a block (e.g., block "A") in EPI line 1 and the location of "the pixel to be interpolated", the location of the block (e.g., block "B") in EPI line 2 is found to measure the MSE as shown Fig. 4. Therefore, grouping the EPI line 2 is not needed. Note that in the block-based matching algorithm, the best-matched block among all blocks in the groups, which are located in the maximum disparity, is chosen for interpolation. However, in the proposed method, we have many candidate blocks equal to number of groups. The number of groups is the truncated value of "B/O".

In Fig. 5, the light grayed block in each group is candidate block of that group. In the next step, the overlapped parts among the candidate blocks are found. In Fig. 5, the overlapped parts are depicted with dark grayed and they are called "candidate pixels" of EPI line 1. Therefore, we could converge the search area in the maximum disparity to limited number of pixels (i.e. candidate pixels), so that we can use the pixel-based matching method in the following of the algorithm, efficiently. Hence, in the following, the proposed algorithm searches for the best matched pixel (we call it "recommended pixel") among the candidate pixels of EPI line 1, to the pixel in EPI line 2. Finding the "recommended pixel" is similar to the pixel-based block matching mentioned in section 4.1. In other words, having the location of a pixel in the candidate pixels of EPI line 1 (e.g., pixel "a") and the location of "the pixel to be interpolated", the location of the pixel in EPI line 2 (e.g., pixel "b") is found to measure the absolute value difference, as shown Fig. 3. Now we have one pixel for interpola-
tion, which is called the recommended pixel.

If the algorithm is stopped in this step, the recommended pixel is used for interpolation. However, the same algorithm from beginning can be done for different "B" and "O" in the assigned maximum disparity. In this case, for each set of parameters ("B", and "O"), we have one recommended pixel. Thus, to find one pixel for interpolation, we find the best matched pixel (we call it the "best-corresponded pixel") among all recommended pixels using the pixel-based method.

Fig. 6 shows the flowchart of the proposed matching method. This flowchart summarized the algorithm of the proposed matching method for ray-space interpolation under the EPI constraint. The searching algorithm is started after choosing the maximum disparity and the location of the "pixel to be interpolated" as input data. For given block and offset sizes, the EPI line 1 is grouped. Then, for each block of a group in EPI line 1, MSE value is calculated by comparing with the corresponded block in EPI line 2 having the location of the "pixel to be interpolated". After calculating MSEs for all blocks in each group, the candidate block is found where has the minimum MSE. Same search is done for all groups. Now, the "candidate pixels" can be determined by finding the overlapped parts among the candidate blocks. The search area is small for candidate pixels area. Therefore, the pixel-based matching method can be used among the candidate pixels to find the "recommended pixel". The same algorithm can be applied to different parameters sets "B", and "O". Finally, the "best-corresponded pixel" pair among all recommended pixels is chosen and applied for interpolation.

6. Experiment

In the experiment, SNR and computational time for ray-space interpolation using the proposed matching method has been compared with the conventional matching methods. The experimental images data are provided by capturing along a line with one camera and 5mm interval. The object has about 30cm distance from the camera. The captured images size is 640 × 480. The interpolated view is the middle view between the first captured image and other captured image, which has the desired distance (i.e., camera interval) to the first image. The SNR of the interpolated image is calculated by using the original captured image that is in the location of the interpolated image.

To evaluate the performance of the proposed method, we use optimum block size throughout the experiments. Fig. 7 shows the SNR for O=1 and the camera interval equal to 100mm when the block size varies from B=2~9 pixels. B=5 or 6 pixels has the best SNR, for all disparities.

Fig. 8 shows that the SNR of the proposed method has been improved in comparison with the conventional matching methods, especially in the high disparities. The block size varies B=3~5 pixels for the block-based,
hierarchical and the proposed matching methods. Offset size is \(O=1\) pixel in the proposed method. The pixel-based method performs well in the small camera intervals. However, when the camera interval is increased its SNR is reduced. Instead, the block-based method performance is less than the pixel-based in the small camera interval, and better in the large camera intervals. The hierarchical matching method performance is better than the block-based method. According to the curves of Fig. 8, the proposed matching method for ray-space interpolation has the best performance among the conventional method, especially for large disparities.

Fig. 9 shows the interpolated images using the pixel-based, block-based, hierarchical and the proposed block matching methods in comparison with the original view. The camera interval is 20mm and the middle image is generated. According to this figure, in the edges area, the proposed method performs well in comparison with the block-based method. The pixel-based method has error, because the disparity is not so small. However in the edge area the performance is good. In other areas, the block-based and the proposed offset-block matching methods have the same performance. The hierarchical block matching method performs well in the edge area. However the result of the proposed method is the most similar to the original image.

The proposed method performance can be evaluated by varying the parameters in the algorithm (i.e., block sizes \(B\) and offset size \(O\)). Fig. 10 shows the interpolation SNR using the proposed matching method for different parameters. This analysis shows that when the algorithm is applied to different offset sizes it will have better decision of the best-corresponded pixels pair for interpolation. Furthermore, the SNR of interpolation for the proposed method when \(O\geq 1\) is still better than conventional matching methods.

Complexity of the offset-block matching method algorithm when it uses for multiple block sizes causes a high computational time for interpolation. If the less computational time for ray-space interpolation is preferred to slight degradation in SNR (according to Fig.
10), then offset sizes "O" more than one pixel is preferred. Fig. 11 shows the computational time of an interpolation using the proposed matching method for the different "B" and "O", the block-based and the hierarchical matching methods with B=5. In this analysis, for the case of O=3 a simplified code was written, and the computational time was compared with the other cases. As a result, the computational time for interpolation using the proposed block matching method when O=3 is 1/6 of the computational time when O=1. However there is a slight degradation of SNR for O=3. The faster computational time of the proposed offset-block matching method for the case of (B=5, O=3) than the block-based and hierarchical matching methods (B=5) makes it suitable to perform a realtime free viewpoint generation of multi-view images in FTV system.

In FTV1), interpolation task is performed in a distributed fashion in the network of computers, which involved with free viewpoint generation. The basic hardware configuration of FTV is depicted in Fig. 12. The experimental system has 16 clients and a server computer. Each computer is a PC cluster that consists of Intel Pentium III 800MHz as CPU with 256Mbyte RAM. Each PC is a general-purpose PC, which has an image capturing board mounted in a PCI bus on each client PC. Gigabit Ethernet connects client PCs with a server PC. Cameras are set with 3-degree (20mm) intervals on an arc-array and about 35cm distance from the object plane. In our system, we use 160 x 120 pixels color image. This system does not need any special hardware such as special computer or depth sensor. The user requests the viewpoint from server side. Table 1 shows the frame rate of the proposed algorithm used in FTV system in comparison with the block-based method. 24.9% improvement has been obtained in frame rate using the proposed method in FTV.

7. Discussion

The block matching methods can generally be evaluated in terms of quality and computational time performances. In this section, in addition to aforementioned terms, the proposed method is discussed according to its compatibility for distributed architecture when realtime operation is required. Hence, the three major advantages of the proposed method are listed in the following.

- **Better quality in the edge area**: Taking the advantages of the block-based and the pixel-based methods simultaneously improves the interpolation quality in the edge area. The experimental results show that about 84% of cases, majority of candidates are overlapped, so that the candidate pixels are less than the half block size "B" in 84% of cases. Therefore, the search area in the pixel-based matching stage of the proposed method is almost small, which gives the condition for better detection of the best correspondence using the pixel-based matching method. Note that when there is not many edge area in the scene, the interpolation quality considering the SNR as evaluation parameter will be almost the same for all matching methods. However, subjective evidence can show the outperforming of the proposed method in the edge area in comparison with the conventional methods.

- **Reduction of Computational cost**: According to the experimental result, the computational cost of the proposed matching method depends on the parameters (i.e., "B" and "O") throughout the algorithm. Furthermore, the degradation of the interpolation quality is small in comparison with the significant reduction on computational time. Hence, this characteristic of the proposed method makes it interesting to be used in real-time and off-time applications.

- **Suitable for distributed processing**: The algorithm of the proposed matching method makes it easy to be used in distributed architecture of the realtime video rendering systems like FTV. In FTV, to accelerate the interpolation time, it is optimal to distribute the block matching task among nodes (i.e., client PCs). In the proposed matching method, we have groups of blocks, and the algorithm chooses one block of each group as candidate block in each client of FTV, after sharing data among clients. In the next step, after gathering all candidate blocks at server, the best corresponding pixels are found for interpolation.

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame Rate Comparison for free viewpoint generation in FTV using block-based and offset-block matching methods (FPS: Frame Per Second)</td>
</tr>
<tr>
<td>Block-based (B=6)</td>
</tr>
<tr>
<td>7.65 FPS</td>
</tr>
</tbody>
</table>

8. Conclusion

In this paper, we have proposed a new block matching method for ray-space interpolation. However, the proposed matching method can be applied to other applications for corresponding search problem with some modifications. The proposed method can also be used for real-time rendering applications, performing in a distributed architecture.

The proposed method has the advantages of the block-based and the pixel-based methods, simultaneously. In other words, the proposed offset-block matching limits the search area for corresponding. This area is small, therefore we can use the pixel-based algorithm to enhance interpolation result. Comparison of the proposed matching method with the conventional matching methods shows its better performance in the edge area for large disparities as well as performing well in small disparities like the pixel-based matching method.

We would like to thank Mr. Purim Na Bangchang sincerely for his helps to evaluate the proposed algorithm practically on FTV system.

[References]


Mehrdad Panahpour Tehrani received the B.E. and M.E. degrees in Electronic Engineering from Tehran Polytechnics University, 1997 and Tarbiat Modarres University-Tehran, Iran 2000, respectively. Currently, he is a candidate for a doctor of engineering at Nagoya University, Japan. His research interests include 3D image processing, and communication in camera sensor network.

Toshiaki Fujii received the B.E., M.E., and Dr. Eng. degrees in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1990, 1992, and 1995, respectively. He is currently an Associate Professor in the Graduate School of Engineering, Nagoya University, Japan. His research interests include 3D image processing and 3D visual communications.

Masayuki Tanimoto received the B.E., M.E. and Dr.E. degrees in electronic engineering from the University of Tokyo, Tokyo, Japan, in 1970, 1972 and 1976, respectively. He has been a Professor of the Department of Information Electronics, Nagoya University, Japan. His current research interests include image coding, multidimensional signal processing, video processing, 3D and HD images, and multimedia systems.