Eye-Contact Visual Communication with Virtual View Synthesis

Yuyu LIU, Keisuke YAMAOKA, Hiroyuki SATO*, Akira NAKAMURA, Yoshiaki IWAI, Ken-ichiro OOI
Weiguo WU and Takayuki YOSHIGAHARA
Information Technologies Laboratories, Sony Corporation, Tokyo, 141-0001 Japan
*Yamagata University, Yonezawa-city, Yamagata, 992-8510 Japan

Abstract—In this paper, we propose a new visual communication system where eye contact is possible by using a virtual image. The virtual image is obtained by view synthesis with stereo matching from two real camera views. We developed a region based Dynamic Programming (DP) approach with improved matching cost, occlusion cost and vertical smoothness constraint. We also proposed a fast view interpolation method. To achieve real time performance, we developed a hardware system. Furthermore, to avoid the reordering problem in the foreground region, a view change approach with disorder detection was adopted. Experiment results demonstrate the validity of our improved DP matching algorithm and eye-contact visual communication system.

Keywords—Eye contact; visual communication; stereo cameras; dynamic programming; virtual view synthesis; real-time system

I. INTRODUCTION

Visual communication is one of the important applications in a broadband network, especially for teleconference which is widely applied in the world. However, eye contact is difficult to realize by a single camera in a visual communication system. This lack of eye contact may degrade the quality of communication [1].

One solution of this problem is to use two cameras to synthesize a virtual view. Aimed at this, some solutions have been developed. Generally, they can be divided into two classes: model-based and image-based. The model-based method tries to build a detailed head model and reproject it to the virtual view [2]. It is limited to the people’s face and thus can not build a complete virtual view, e.g. where a person’s hand is included in the virtual view.

Image-based methods [3-5] are more general. They can build complete virtual views. However, they require the knowledge of the correspondences between pixels of the two images. Building correspondences, also called stereo matching, is critical for virtual view synthesis. The precision directly decides the quality of the virtual view. If they are matched successfully, we can employ methods like view interpolation to synthesize the virtual view.

There are many methods to perform stereo matching, such as Winner-take-all (WTA), Scanline Optimization (SO), Dynamic Programming (DP), Graph Cut (GC), etc [6-11,13]. Among them, WTA is a local minimum approach. It finds correspondences between pixels by correlating surrounding windows. SO and DP can find a global minimum in one scanline and hence produce better results. The powerful GC method can reach close to the global minimum (up to a coefficient) of an energy function. Unfortunately, we found that none of the stereo matching algorithms above can give satisfied performance in a complicated environment with a large disparity range. In a visual communication application where cameras are placed on the either side of a screen, disparity may be large.

To solve the eye contact problem based on today’s technologies, we developed the region based DP algorithm for creating a virtual view from two real images. We adopt DP based on two reasons. First, compared to other methods, DP produces a more detailed disparity map in our application. Thus it potentially provides a better virtual view. Second, DP has relatively low algorithm complexity and is therefore convenient to realize in hardware.

In our proposed system, we first segment out foreground with people and perform processing on foreground only. This segmentation greatly reduces the possibility of matching failures and guarantees the quality of virtual view. Second, we propose a region based DP approach with improved matching cost, occlusion cost and vertical smoothness constraint. Third, we develop an approach to solve the reordering problem. When the ordering constraint is violated in the segmented region, DP may fail to find the correct matching. We propose a view change technique with disorder detection to solve this problem. Finally, we develop an eye-contact visual communication system with a real-time hardware unit for our improved DP matching algorithm and fast virtual view synthesis. The validity of our improved DP matching algorithm is shown. Experiment results demonstrate that people using our eye contact visual communication system.
The paper is organized as follows. Section 2 shows our eye-contact visual communication system with a real-time hardware unit. Section 3 introduces the improved DP matching approach. Section 4 describes our approach for virtual view synthesis including our solution to the reordering problem. Experimental results are shown in Section 5.

II. EYE CONTACT SYSTEM FOR VISUAL COMMUNICATION

A. System overview

Our system is shown in Fig. 1. Two cameras are placed at the left and right sides of a screen. From images taken by both cameras, the virtual view-synthesis hardware builds a virtual image as it would have been taken by a hypothetical camera in the center of the screen. The virtual image is transferred via network and displayed on the screen on the other side. When both sides use our system, their eyes can successfully meet each other.

B. Real-time hardware unit for virtual view synthesis

Considering complexity of image rectification and DP matching, it is hard to achieve real-time performance when using standard PC hardware only. Therefore, we developed the virtual view synthesis hardware, as shown in Fig. 2.

For processing, images are first rectified according to the camera parameters table [14,15]. Secondly, foreground regions are segmented out. Then our improved DP matching algorithm is performed in the segmented regions. Finally, the virtual image is synthesized. Hardware parameters of our system are described in Table 1.

Table 1 Hardware parameters

<table>
<thead>
<tr>
<th>Input</th>
<th>Signal</th>
<th>NTSC, RGB, Y/C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Size</td>
<td>640x480</td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>Signal</td>
<td>NTSC, Y/C, IEEE 1394</td>
</tr>
<tr>
<td>Image Size</td>
<td>640x480</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Latency</td>
<td>1 frame</td>
</tr>
<tr>
<td>Speed</td>
<td></td>
<td>30 frame/s, 60 field/s</td>
</tr>
</tbody>
</table>

III. REGION BASED DP APPROACH

A. Traditional method

As described in [6,7], traditional DP performs stereo matching scanline by scanline between left and right images, as shown in Fig. 4a. The physical cause of each possible jump is illustrated in Fig. 4b.

For processing, images are first rectified according to the camera parameters table [14,15]. Secondly, foreground regions are segmented out. Then our improved DP matching algorithm is performed in the segmented regions. Finally, the virtual image is synthesized. Hardware parameters of our system are described in Table 1.

Table 1 Hardware parameters

<table>
<thead>
<tr>
<th>Input</th>
<th>Signal</th>
<th>NTSC, RGB, Y/C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Size</td>
<td>640x480</td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>Signal</td>
<td>NTSC, Y/C, IEEE 1394</td>
</tr>
<tr>
<td>Image Size</td>
<td>640x480</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Latency</td>
<td>1 frame</td>
</tr>
<tr>
<td>Speed</td>
<td></td>
<td>30 frame/s, 60 field/s</td>
</tr>
</tbody>
</table>

III. REGION BASED DP APPROACH

A. Traditional method

As described in [6,7], traditional DP performs stereo matching scanline by scanline between left and right images, as shown in Fig. 4a. The physical cause of each possible jump is illustrated in Fig. 4b.

For processing, images are first rectified according to the camera parameters table [14,15]. Secondly, foreground regions are segmented out. Then our improved DP matching algorithm is performed in the segmented regions. Finally, the virtual image is synthesized. Hardware parameters of our system are described in Table 1.

Table 1 Hardware parameters

<table>
<thead>
<tr>
<th>Input</th>
<th>Signal</th>
<th>NTSC, RGB, Y/C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Size</td>
<td>640x480</td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>Signal</td>
<td>NTSC, Y/C, IEEE 1394</td>
</tr>
<tr>
<td>Image Size</td>
<td>640x480</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Latency</td>
<td>1 frame</td>
</tr>
<tr>
<td>Speed</td>
<td></td>
<td>30 frame/s, 60 field/s</td>
</tr>
</tbody>
</table>

III. REGION BASED DP APPROACH

A. Traditional method

As described in [6,7], traditional DP performs stereo matching scanline by scanline between left and right images, as shown in Fig. 4a. The physical cause of each possible jump is illustrated in Fig. 4b.

For processing, images are first rectified according to the camera parameters table [14,15]. Secondly, foreground regions are segmented out. Then our improved DP matching algorithm is performed in the segmented regions. Finally, the virtual image is synthesized. Hardware parameters of our system are described in Table 1.

Table 1 Hardware parameters

<table>
<thead>
<tr>
<th>Input</th>
<th>Signal</th>
<th>NTSC, RGB, Y/C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Size</td>
<td>640x480</td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>Signal</td>
<td>NTSC, Y/C, IEEE 1394</td>
</tr>
<tr>
<td>Image Size</td>
<td>640x480</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Latency</td>
<td>1 frame</td>
</tr>
<tr>
<td>Speed</td>
<td></td>
<td>30 frame/s, 60 field/s</td>
</tr>
</tbody>
</table>

III. REGION BASED DP APPROACH

A. Traditional method

As described in [6,7], traditional DP performs stereo matching scanline by scanline between left and right images, as shown in Fig. 4a. The physical cause of each possible jump is illustrated in Fig. 4b.

For processing, images are first rectified according to the camera parameters table [14,15]. Secondly, foreground regions are segmented out. Then our improved DP matching algorithm is performed in the segmented regions. Finally, the virtual image is synthesized. Hardware parameters of our system are described in Table 1.

Table 1 Hardware parameters

<table>
<thead>
<tr>
<th>Input</th>
<th>Signal</th>
<th>NTSC, RGB, Y/C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Size</td>
<td>640x480</td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>Signal</td>
<td>NTSC, Y/C, IEEE 1394</td>
</tr>
<tr>
<td>Image Size</td>
<td>640x480</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Latency</td>
<td>1 frame</td>
</tr>
<tr>
<td>Speed</td>
<td></td>
<td>30 frame/s, 60 field/s</td>
</tr>
</tbody>
</table>

III. REGION BASED DP APPROACH

A. Traditional method

As described in [6,7], traditional DP performs stereo matching scanline by scanline between left and right images, as shown in Fig. 4a. The physical cause of each possible jump is illustrated in Fig. 4b.

For processing, images are first rectified according to the camera parameters table [14,15]. Secondly, foreground regions are segmented out. Then our improved DP matching algorithm is performed in the segmented regions. Finally, the virtual image is synthesized. Hardware parameters of our system are described in Table 1.

Table 1 Hardware parameters

<table>
<thead>
<tr>
<th>Input</th>
<th>Signal</th>
<th>NTSC, RGB, Y/C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Size</td>
<td>640x480</td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>Signal</td>
<td>NTSC, Y/C, IEEE 1394</td>
</tr>
<tr>
<td>Image Size</td>
<td>640x480</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Latency</td>
<td>1 frame</td>
</tr>
<tr>
<td>Speed</td>
<td></td>
<td>30 frame/s, 60 field/s</td>
</tr>
</tbody>
</table>

III. REGION BASED DP APPROACH

A. Traditional method

As described in [6,7], traditional DP performs stereo matching scanline by scanline between left and right images, as shown in Fig. 4a. The physical cause of each possible jump is illustrated in Fig. 4b.
match pair \((X_L, X_R)\). In the second step, DP backtracks over the matrix of match pairs depending on the search results to find the optimal matching path.

It is well known that DP suffers from the following two drawbacks [7]. First, it treats each scanline independently. Hence, it is hard to guarantee the alignment between scanlines, which may degrade the overall quality greatly. Second, DP always searches forward, which enforces ordering constraint originally, though this constraint is not always satisfied in reality. Therefore, we improve the traditional DP to make it capable for our application.

**B. Region Based DP**

We employ a region based DP because of the four reasons below. First, a complex background environment easily violates the ordering constraint. Second, when disparity changes abruptly, DP will be too sensitive to the \(\text{OccCost}\). Third, synthesizing virtual view between foreground and background tends to produce “haloing” as mentioned in [3]. Finally, and most importantly, in visual communication we are concerned with people only. It is unnecessary to include the background for “eye contact”.

Hence, before matching, we initially perform foreground region segmentation to cut out people. After performing DP matching and virtual view synthesis on segmented regions, we place it back into the background scene observed by the real cameras.

**C. Improved Matching Cost**

Although region based DP makes matching simpler, it is still necessary to improve the DP algorithm to raise the quality of the virtual image. First and most intuitive one is the modification of matching cost. As mentioned above, there are many choices. For example, reference [12] brought out a sampling insensitive cost. And reference [3] furthermore used the window correlation. But, all of these choices only use image intensities. Generally speaking, intensity is helpful to match detailed parts of stereo images. But color information can be helpful for matching large areas. So we integrate color information to form a new cost. It does outperform the cost with intensity only. Fig. 5a shows the synthesized virtual view by using intensity only. Fig. 5b is the result by using both intensity and color information. We can see that wrong matches in red circle have been corrected.

Before calculating, we first segment two features from YUV values (For RGB images, a transformation should be performed first), as shown in (2).

\[
\begin{align*}
F_1 &= Y \\
F_2 &= U - V
\end{align*}
\]  

(2)

\(F_1\) Corresponds to intensity. And \(F_2\) holds the key information of color. Then, matching cost of current scanline \(y\) can be calculated as in (3).

\[Cst(X_L, X_R; y) = k \cdot |F_1(X_L; y) - F_1(X_R; y)| + \]
\[\quad (1 - k) \cdot |F_2(X_L; y) - F_2(X_R; y)|
\]  

(3)

Here \(Cst(X_L, X_R; y)\) is the matching cost of pair \((X_L, X_R)\) in scanline \(y\). \(k\) is a parameter to adjust the rate between intensity and color information. In our system, we set \(k=0.5\).

Integrating color information can only improve intra-scanline DP matching. However, as mentioned above, inter-scanline misalign errors may degrade virtual image quality greatly, too. Aimed at this defect, Reference [6] used inter-scanline DP search to correct it. Reference [3] performed Gaussian smoothing filter on cost volume. But we found this error can be improved by simply summing weighted matching cost of its neighbor scanlines with current scanline. So our final matching cost is computed as (4).

\[M(X_L, X_R; y) = \sum_{i=-m}^{m} w_i \cdot Cst(X_L, X_R; y + i)
\]  

(4)

\(w_i\) weight the different scanlines and sum up to one, \(\sum w_i = 1\). Generally, current scanline should have a bigger weight. \(m\) is the parameter to control the range of relationship calculated in neighbor scanlines. If \(m\) is too large, disparity of current scanline and its neighbors may be different. This will on the contrary degrade the performance. Hence, in our system, we set \(m=1\) and \(w_0 = 0.4, w_i = 0.3\).

**D. Improved Occlusion Cost**

In large areas exhibiting low texture, DP becomes very sensitive to the \(\text{OccCost}\) and tends to make errors. Correcting such kind of error is especially important for our system because, after region segmentation, large background areas will be left blank. Of course we can choose only to match foreground regions. But, it will be hard for hardware implementation.

We solve this problem by modifying the DP search process. The improved one can be described by (5).

\[C(X_L, X_R) = \min \left\{ \begin{array}{l}
C(X_L - 1, X_R) + \text{OccCost} + \alpha \cdot M(X_L, X_R; y) \\
C(X_L - 1, X_R - 1) + M(X_L, X_R; y) \\
C(X_L, X_R - 1) + \text{OccCost} + \alpha \cdot M(X_L, X_R; y)
\end{array} \right. \]  

(5)

where \(\alpha\) is a constant satisfying \(0 < \alpha < 1\). By adding \(\alpha \cdot M(X_L, X_R; y)\) as a penalty to occlusion cost, we can
increase the possibility for DP to declare two pixels as matched so as to improve its performance in low texture or blank areas. In our experiments, we choose $\alpha = 0.25$ for all the images. We found the setting of $\alpha$ is not sensitive.

E. Matching Error Improvement

In some scanline, DP can be dragged by some area and fail in other important areas, as shown in Fig. 6a. Such kind of error happens rarely and, generally, not scanline continuous.

![a. Error](image1) ![b. Detection](image2) ![c. Corrected](image3)  
**Fig. 6 Scanline error and correction**

According to this observation, through comparing among neighbor scanlines as in Fig. 6b, we can detect this error using a median filter in the vertical direction. Furthermore, we use reliable disparity values in the previous scanline to replace disparity values of current scanline. Then, we can modify the correspondences by the corrected disparity map, as in Fig. 6c.

![a. Error](image4) ![b. Detection](image5) ![c. Corrected](image6)  
**Fig. 6 Virtual image improvement**

The improvement of the virtual image for this error correction is shown in Fig. 7.

IV. VIRTUAL VIEW SYNTHESIS

A. Virtual View Synthesis Approach

Given the results of DP search, we use a projection model to build the new virtual view. The projection model is shown in Fig. 8a.

![a. Middle view](image7) ![b. Multiple view](image8)  
**Fig. 8 Virtual View Projection Model**

The model demonstrates the view projection of one scanline. Because DP is performed scanline by scanline, the virtual view is synthesized line by line, too.

DP search results in an optimum path from end point to start. We found that, in fact, we just need to project this path to the virtual image plane. Considering the physical meaning of each DP jump, we perform three operations: copying a pixel from left, copying a pixel from right and mixing two pixels from both left and right. The three operations correspond to occlusion to right, occlusion to left and matched respectively.

As shown in Fig. 8a, the projection will cause the number of pixels to be doubled. So we first upsample the virtual view to its double. Then we perform the projection. Finally, we downsample the virtual view to its original size.

This approach can be perfectly merged into the backtracking process of DP. Hence we can perform DP backtracking and virtual view synthesis in one pass only. Since upampling and downsampling can be performed pixel by pixel, memory and computation can be further saved following a simple hardware implementation.

Furthermore, when we extend the projection plane to other chords of the circumscribed circle, we can compute in parallel any virtual view placed between the two cameras. This is very important for our view change function afterward.

B. Human Region Segmentation

We use background subtraction to segment out human regions. For this aim, we initially take two background (left and right) scenes without any foreground objects present. Since background subtraction often leaves many fragments, we employ median filtering and a connected component labeling process to further revise it.

C. Disorder Detection

Although region based DP matching is more robust, ordering constraint still may be violated, e.g. when the person stretches out a hand. In Fig. 9a and Fig. 9c we can see the order of hand and neck is reversed. Such situations occasionally appear in several frames. To distinguish this from the violation of the general ordering constraint, we call it the reordering problem. When it happens, DP is unable to compute a correct virtual view, as shown in Fig. 9b.

![a. Left Image](image9) ![b. Virtual Image](image10) ![c. Right Image](image11)  
**Fig. 9 Disordered view interpolation**

In order to solve it, we first use a disorder detection which decides whether DP performs the correct matching. The detection can be divided into two processes: First we rematch the stereo images by some other stereo matching algorithm which is not refrained to the ordering constraint. Then we compare the two disparity maps to find out whether a disorder exists.
To perform comparison, the additional disparity map can be sparse but must be as reliable as possible. Considering the simplicity for hardware implementation, we developed the Symmetric Shifted Multiple Window (SSMW) algorithm. It is some kind of area based matching algorithm. In it, we combined the shifted window [8] and multiple windows [9] techniques successfully, as demonstrated in Fig. 10. In our system, we use the same 9 shifted windows as in [8] and 3 multiple windows, sized at 3×3, 5×5 and 9×9.

SSMW uses local windows to match pixels and is not refrained to the ordering constraint. Since we use methods such as reliable point verification, shifted multiple window matching, match reliable verification and so on, SSMW is able to build a reliable disparity map as dense as possible. We call it symmetric since unique constraint is enforced, which guarantees the result matched left to right is same as the one matched right to left.

It is worth to mention that, unlike DP, SSMW employ image intensity ($F_1$) only. Experiments show that costs using $F_1$ only outperform costs using both, intensity and color. This can be explained by the locality of SSMW. Color ($F_2$) helps to match large areas but contradictorily degrades performance locally.

Based on the disparity maps produced by DP and SSMW, a comparison algorithm is developed to decide whether DP works well. First, histogram processing is performed on the disparity map produced by SSMW. By checking envelop of it, we can detect whether reordering problem may exist. Then we employ a shifted window to compare the two disparity maps so as to decide whether a disorder exists.

### D. View Change

When re-ordering problem happens and DP fails, we use view change to avoid disordered virtual image. View change is a step by step process. When disorder is initially detected, we begin to switch the position of virtual view from the center to one of the real cameras. Since there is no disorder in real images, the virtual view becomes consistent again after a view change has been completed.

One possible view change process is shown in Fig. 11. In frame $t$, disorder is initially detected. Then view change process is activated. Frame by frame, virtual view is switched from middle to the right step by step. In our system, the step size is $B/8$, where $B$ is the baseline between cameras. So, after 4 frames in frame $t+4$, view change is finished. And virtual view comes same as the real view taken by the right camera.

Both, in and after view change, DP matching and disorder detection are still running. When detection claims no disorder, a reverse view change process happens, which shifts the virtual view from real right view back to the middle frame by frame. When the reverse process is finished, virtual view returns back to the center. Then eyes can contact again.

### E. Noise Removing in Occluded Region

Occlusions are areas which can be seen in only one of the stereo views. When performing virtual view synthesis, we have to interpolate those pixel values from only one view. Since the same 3D point may have different pixel values in the two views, interpolation cause them look strange in the virtual image when compared to the interpolated pixels in non-occluded regions. Especially, when specularity occurs in one image, this difference can be very large.

To improve the quality of the virtual view, we use a smoothing filter to reduce the noise in occlusion regions. In the first step, we copy the pixel values as before and mask the position. Next, in all the masked positions, we perform a median filter on the virtual image. By such a method, we can make these areas look more natural.

### V. EXPERIMENT RESULTS

We perform experiments in an office environment. To testify the validity of our method, experiments cover situations
of both single and multiple persons. In experiments, the image size we processed is \(640 \times 480\). The baseline between two cameras is 32cm. And the distance from people to camera is about 65cm.

A. Single person

When only one person exists, simulation results of our system are as shown in Fig. 12. In the complex office environment, our method can well segment out the human region. After virtual view is successfully built, we assemble background back. Finally we get a natural center view where eye can contact.

B. Multiple persons

When multiple persons exist, by region segmentation, we can cut out the human region which we want to make eye contact and perform virtual view synthesis. Fig. 13 shows the simulation results. The person in the middle is guaranteed for obtaining eye contact.

VI. Conclusions

This paper brought out a system to make eye contact in visual communication applications. By region segmentation and DP improvement, it raises the quality of virtual view greatly. Our improvement of DP is also applicable to other stereo matching areas, especially for large disparity range. We also find out a fast method to perform view interpolation based on DP matching. To realize processing in real time, we designed a hardware system. Through it, image rectification, region based DP matching and virtual view synthesis can be finished at the speed of 30 frames per second.

Considering the reordering problem, we presented a way to detect whether disorder exists. When detected, we propose the view change method to avoid the disordered virtual view. Finally, we revise the view interpolation in occlusion regions to make virtual view look more natural.

In future work, we will investigate how to improve our matching algorithm in order to solve the reordering problem directly without the need of changing the viewing position.

ACKNOWLEDGMENT

The authors would like to thank Naohisa Ohta, Yusuke Sakai, Atsushi Yokoyama, Kazuyoshi Hayashi, Hiroshi Hayashi, Masashi Takao, Akihiro Kikuchi, Yuichi Motomura, Koichi Uchide and Takao Hiramoto of Sony Corporation for valuable discussion and useful comments on issues related to this paper.

REFERENCE