VIEW SYNTHESIS BASED ON CONDITIONAL RANDOM FIELDS AND GRAPH CUTS

Lam C. Tran\textsuperscript{1}, Christopher J. Pal\textsuperscript{2}, and Truong Q. Nguyen\textsuperscript{1}

\textsuperscript{1} ECE Department, University of California San Diego, La Jolla CA 92093
\textsuperscript{2}École Polytechnique de Montréal, Montréal, QC, Canada.

ABSTRACT

We propose a novel method to synthesize intermediate views from two stereo images and disparity maps that is robust to errors in disparity maps. The proposed method computes a placement matrix from each disparity map that can be used to correct errors when warping pixels from reference view to virtual view. The second contribution is a new hole filling method that uses depth, edge, and segmentation information to aid the process of filling disoccluded pixels. The proposed method selects pixels from segmented regions that are connected to the disoccluded region as candidates to fill the disoccluded pixels. We also provide an explicit probabilistic model to select the best candidate for each disoccluded pixel efficiently with Conditional Random Fields (CRFs) and graph-cuts.

Index Terms— Autostereoscopic Display, CRF, Disparity Map, Graph-Cuts, Stereo Images, and View Synthesis.

1. INTRODUCTION

The availability of autostereoscopic displays as consumer electronics has sparked a growing interest in developing algorithms to adapt existing content for viewing on these displays. The autostereoscopic display enables 3D viewing without the need of polarized glasses and offers free viewpoint for a given scene. These displays, compared to the conventional stereoscopic displays, offer a more natural 3D viewing experience and are expected to be one of the next generation displays for home and industry. However, existing stereo content must be adapted for this type of display. A solution to this problem is to synthesized multiple stereo images for each pair of stereo images and disparity maps.

Depth image-based rendering (DIBR) is commonly used to generate new virtual viewpoints for autostereoscopic displays. The three main steps of the DIBR framework are: preprocessing of disparity maps, image warping, and hole filling. The challenge of this method is the hole filling step in which one must restore the occluded pixels in the new virtual view. Disocclusion refers to the process of recovering scene information obstructed by visible points and we refer to any occluded pixels that have been restored as disoccluded pixels. In [1], a Gaussian filter is used to smooth the disparity map in the preprocessing step to eliminate disocclusion pixels. This method is easy to implement and computationally efficient; however, the synthesized image is unrealistic due to geometry distortion especially when the disoccluded region is large.

A better approach [2] combines depth-based-hole-filling and in-painting to restore the disocclusion pixels more accurately compared to prior in-painting methods without using depth information. While in-painting is a powerful tool to restore small disoccluded regions, it produces a notable blur and can become computationally inefficient when the disoccluded region is large in the virtual view. Both of these methods described above produce visual artifacts shown in [3] and degrade the 3D effect when the synthesized images are interlaced into a multiview 3D image.

In this work, we propose a new method to synthesize intermediate stereo images from a pair of stereo images by using depth, edge, and image segmentation as features in a probabilistic framework to achieve high accuracy in both subjective and quantitative metrics. Edge, depth, and segmentation features will provide more information to correctly fill in the disoccluded pixels and reduce visual artifacts. In this work, we use a lattice structured CRF and graph cuts minimization framework proposed in [4, 5] for stereo vision. Here, the formulation is changed with the data and smoothness terms enforcing the labeling of each disoccluded pixel to agree with neighboring pixels and regions connected to the disoccluded region respectively. The goal is to generate new virtual views between the two reference images to create content for the autostereoscopic display.

The remainder of this paper is organized as follows. Section 2 describes the type of pixels in the virtual view. Section 3 describes how to generate the initial synthesized image and placement matrix. Section 4 describes pixels selection being used to label disoccluded pixels. Section 5 describes our probabilistic model. Sections 6 shows the simulation and experiment results, and section 7 concludes the paper.

2. PIXEL CLASSIFICATION

Each pixel in the virtual view is classified as stable, unstable, and disoccluded. Stable pixels have only one pixel candidate and remain constant throughout the inference process. Un-
stable pixels have multiple pixel candidates and we want to predict the best candidate that minimizes the energy function described in section 5. Finally, disoccluded pixels have no pixel candidate and are occluded in both reference images. The candidates are obtained in regions described in section 4. We use the stable pixels to find the best configuration of the unstable and disoccluded pixels.

3. SYNTHESIS INITIAL VIRTUAL VIEW

In the initial step, color segments in the left \( I_L \) and right \( I_R \) reference images are extracted by [6]. In each segment, pixels with disparity that exceed \( \pm 20 \) from the mode are labelled as occluded, and the occluded pixels are then filled with the disparity of the nearest non-occluded neighboring pixels in the segment to generate left \( D_L \) and right \( D_r \) refined disparity maps. After the initial step, the predefined virtual camera position \( \theta \in \{0, 1\} \) is used to compute two disparity maps as

\[
D_L = \theta D_I \text{ and } D_R = (1 - \theta) D_I.
\]

These disparity maps are used to generate placement matrices to warp pixels in reference images to the virtual view.

3.1. Placement Matrix

We define a placement matrix \( P \) as a sparse matrix that contains 0 or 1 for each element, and the element \( P_{i,j} \) is set to 1 to indicate that pixel \( i \) in the reference view is placed in pixel \( j \) in the virtual view. To take advantage of parallel programming we designed our algorithm to process each row of the reference images separately, and the indices \( i \) and \( j \) become the pixels of a row in the reference image and virtual view respectively.

The placement matrix uses to warp pixels from the right reference image to the virtual view is computed as

\[
P_{R, i, j} = \begin{cases} 1 & \text{if } T_R(\alpha) > 0 \\ 0 & \text{otherwise} \end{cases},
\]

where

\[
i = \alpha - d_{r,\alpha} - 1, \quad j = \alpha, \quad T_R(\alpha) = n \times i + j,
\]

\( n \) is the width of the image, \( d_r \) is a row of \( D_R \), \( \alpha \) is used to index \( d_r \), and \( T_R(\alpha) \) is a function used to check if the mapping is valid. Equation (2) is a mapping of pixel coordinate from the reference image to virtual view, where the first and second equations in (3) calculate the row \( i \) and column \( j \) of \( P_R \) respectively. In the case where a column of \( P_R \) has more than one candidates, the element with the largest \( j \) represents the pixel closest to the camera is kept.

Similarity, the placement matrix uses to warp pixels from the left reference image to the virtual view is computed as

\[
P_{L, i, j} = \begin{cases} 1 & \text{if } T_L(\alpha) \leq n^2 \\ 0 & \text{otherwise} \end{cases},
\]

where

\[
i = \alpha + d_{l,\alpha} - 1, \quad j = \alpha, \quad T_L(\alpha) = n \times i + j,
\]

and \( d_l \) is a row of \( D_L \). However, we keep the smallest \( j \) when there are more than one candidates.

By using two median filters to refine the placement matrix, the proposed method is robust to errors in the disparity map. The median filter of size 3 is used to fill in missing pixel of size one and another filter of size 5 is used to correct placement error in the virtual view. Fig. 1 shows the synthesized images generated by the placement matrices with and without refinement. The synthesized image with the refined placement matrix removes most of the disparity and placement errors.

3.2. Synthesis Virtual View With Placement Matrices

Placement matrices for the stable pixels in the virtual view \( P^1_L \) and \( P^1_R \) are obtained where the column sum of \((P_L + P_R)\) is 1, while the unstable pixels placement matrices \( P^2_L \) and \( P^2_R \) are obtained where the column sum of \((P_L + P_R)\) is 2 and the pixel that is closer to the camera is selected initially. We use the placement matrices discussed above to generate the virtual view as

\[
I_S = I_L(P^1_L + P^2_L) + I_R(P^1_R + P^2_R).
\]

The initial synthesized view is refined with the unstable pixels to enforce spatial consistency of luminance and account for errors in the disparity maps that are not detected in the initial refinement step. We use a discriminative CRF model described in section 5 and graph-cuts to find the best pixel for each unstable pixel. Fig 2 shows the initial and refined synthesized views. The luminance is consistent and the pixels wrapped incorrectly are removed in the refined view.

4. PIXELS SELECTION FOR DISOCCCLUDED REGION

To fill the disoccluded regions, we use [6] to segment the refined virtual view. For each disoccluded region, the algorithm
searches for regions that are connected to the disoccluded region and uses the pixels in the connected regions as pixel candidates to fill the disoccluded pixels as shown in Fig. 3. After the pixel candidates are selected, an optimization method described in section 5 is used to find the best pixel to fill each disoccluded pixel. To take advantage of parallel programming, this algorithm is developed to process each disoccluded region separately.

5. CRF FOR HOLE FILLING

For the unstable and disoccluded pixels, we construct conditional random fields for pixel candidates \( C = \{ c \} \), disparities \( D = \{ d \} \), gradients \( G = \{ g \} \), observed pixels \( \mathcal{I} = \{ i \} \), disocclusion indicators \( O = \{ o \} \), and segments \( S = \{ s \} \) with the following form

\[
P(C|D,G,I,O,S) = \frac{1}{Z(D,G,I,O,S)} \prod_{p \in \mathcal{P}} \Phi(c_p, d_p, g_p, o_p, i_p) \prod_{n \in \mathcal{N}_S} \Psi(c_p, s_n),
\]

where \( \mathcal{P} \) is the number of occluded or unstable pixels, \( \mathcal{N}_S \) is the number of regions connected to the disoccluded region, and \( Z(D,G,I,O,S) \) is the partition function that normalizes the probability of (7) between 0 and 1. The potential function \( \Phi \) enforces the candidate pixel selected to agree with its neighbor pixels, while \( \Psi \) jointly models consistency between neighbor regions.

The negative log of the probability model (7) is computed to yield the data term for selecting candidate \( c \) at pixel \( p \)

\[
U(c_p, d_p, g_p, o_p, i_p) = d_{p,c} \sum_{q \in \mathcal{N}_p} (1 - y_{p,q})(1 - o_{p,q}) |c_p - i_{p,q}|^2 ,
\]

where \( \mathcal{N}_p \) are the neighboring pixels of \( p \), \( d_{p,c} \) is the disparity of candidate \( c_p \), and \( y_{p,q} \), \( o_{p,q} \) are an edge indicator between pixels \( p \) and \( q \), and \( i_{p,q} \) and \( y_{p,q} \) are the intensity value and disocclusion indicator of pixel \( q \) respectively. Neighboring pixel that is disoccluded or has an edge between the disoccluded pixel \( p \) does not influence the candidate selection process. Furthermore, the model is penalized more for selecting a candidate pixel that is closer to the camera since objects farther from the camera is more likely to be occluded than closer objects. The term

\[
V(c_p, s_n) = \min_{r \in S_n} \sqrt{(p(x) - r(x))^2 + (p(y) - r(y))^2}
\]

is the shortest distance to the region \( s_n \) that contains candidate pixel \( c \). The model is penalized more for selecting a candidate pixel in regions that is farther away from the location of the disoccluded pixel.

The goal is to minimize

\[
- \log(P(C|D,G,I,O,S)) = \log(Z(D,G,I,O,S)) + \sum_{p \in \mathcal{P}} U(c_p, d_p, g_p, o_p, i_p) + \sum_{n \in \mathcal{N}} V(c_p, s_n).
\]

We use the winner-take-all approach and a fast alpha-expansion graph cuts algorithm [7] to infer the final synthesized image. For the unstable pixels, the smoothness term is disabled and the disparity coefficient is set to 1 in the data term.

6. EXPERIMENT AND RESULTS

In our experiment, the images from the Middlebury data set [4] are used to evaluate the proposed method. This data set consists of two ground truth disparity maps and 7 stereo images for each data set. Views 1 and 5 are used with the disparity maps to synthesize virtual view 3. These synthesized images are compared with the ground truth to quantify the algorithm performance based on PSNR and SSIM[8] shown in Table 1. The SSIM index value 1 is only reachable when two images are identical and a higher PSNR normally indicates that the synthesized image is of higher quality. Fig. 4 shows the hole filling of the disocclusion regions of the proposed method compared with the ground truth.

The experimental results show that the proposed method achieved on average over 34 dB in PSNR and 0.95 index value in SSIM on the Middlebury stereo data sets. The results reported by [2, 3] are both from different stereo data sets and their experiments are structured differently. We report their
Table 1: Experiment results for Middlebury Stereo Database.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>PSNR</th>
<th>SSIM</th>
<th>Data Set</th>
<th>PSNR</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Art</td>
<td>32.66</td>
<td>0.95</td>
<td>Moebius</td>
<td>34.30</td>
<td>0.94</td>
</tr>
<tr>
<td>Books</td>
<td>30.92</td>
<td>0.93</td>
<td>Monopoly</td>
<td>32.19</td>
<td>0.95</td>
</tr>
<tr>
<td>Dolls</td>
<td>35.99</td>
<td>0.97</td>
<td>Reindeer</td>
<td>33.70</td>
<td>0.94</td>
</tr>
<tr>
<td>Laundry</td>
<td>32.13</td>
<td>0.95</td>
<td>Plastic</td>
<td>37.77</td>
<td>0.98</td>
</tr>
<tr>
<td>Wood</td>
<td>37.47</td>
<td>0.94</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Experiment results comparison.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR (dB)</td>
<td>32.12</td>
<td>29.76</td>
<td>34.02</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.86</td>
<td>Not Reported</td>
<td>0.95</td>
</tr>
</tbody>
</table>

findings in average PSNR and SSIM as an indirect comparison with our framework in Table 2. The results indicate that the proposed method is highly accurate in both subjective and quantitative. Due to space limitation, only portion of the experimental results are shown in this paper. The full size images and full experiment can be viewed or downloaded on our website (http://videoprocessing.ucsd.edu/~lamtran/icip2010.html).

7. CONCLUSION

In this paper, we have proposed a framework to synthesize intermediate views that are robust to errors in disparity maps. The proposed method uses placement matrices to generate an initial virtual view from the reference images and disparity maps. The disoccluded regions in the initial view are then filled with a novel hole filling method that uses depth, edge, and segmentation information. From the experimental result, the proposed framework has great potential for stereo to 3D multi-view conversion.

8. REFERENCES


