Total variation regularization can be used to decompose any natural image into a structure image (edges and smooth areas) and a texture image (texture only). Although structure images can be highly compressed by conventional JPEG methods, texture images require many bits to represent their feature details. Texture images are composed of several repeated patterns, each of which can be synthesized from a small texture image, and the borders of different texture patterns correspond to the edges in a structure image. We present a novel image compression method for these images based on this observation. First, we apply a context-aware resizing method to the input image to obtain a compaction image that has as many unique texture patterns as possible. Then the compaction image is divided into texture and structure images. Our proposed encoder sends the compressed compaction texture and compaction structure images and the compressed structure image extracted from the input image to the decoder side. At the decoder side, a texture image is synthesized from the compaction texture image through matching between the compaction structure image and the original-size structure image. Experimental results show that the decoded image obtained by our proposed method is subjectively similar to the original one, with higher texture feature accuracy than that obtained by a conventional JPEG method but almost the same data size.

Index Terms— Texture synthesis, Total Variation, Bidirectional Similarity

1. INTRODUCTION

In general, natural images consist of smooth regions, edges, and textures. Edges are a combination of high and low frequency components, while textures mainly consist of high frequency components. When we compress a natural image by a conventional JPEG encoder, which reduces the higher frequency components first, the result is a ringing artifact on the edges and a lack of textures in the decoded image. The ringing artifact can be removed by using an optimization decoding scheme based on total variation (TV) prior [2]. However, recovering the lost texture is impossible in this framework.

TV-regularization [1], on the other hand, can be used to decompose natural images into structure images that consist of flat regions and edges, and texture images that consist of textures. The structure images do not contain any textures, and they can be highly compressed. The textures are composed of several repeating patterns. Previous studies [3][4] utilized this repeating patterns feature for image compression. However, the difference between these works and ours is that our method does not send the explicit mapping of the texture patterns to the decoder.

We propose an image-coding framework based on these observations that uses two different encoding schemes for the two decomposed image components. We use a conventional JPEG encoder and optimization decoder [2] for the structure image. For the texture image, by using the texture synthesis described in Section 3, we developed an effective texture compression method that achieves high subjective quality. At the decoder side, the final decoded image is obtained by combining the decoded structure image and texture image. First, we describe the techniques that our method use in Section 2.

2. PREVIOUS WORK

2.1. TV REGULARIZATION

Rudin et al. proposed TV-regularization for image denoising [1]. The TV norm is the summation of the absolute difference between the target pixel and the surrounding pixels. Thus, the TV norm of image $u$ is defined by

$$ TV(u) = \int |\nabla u| dx $$

where $\nabla u$ is the weak gradient of $u$. Images with small TV norms are considered to be smooth images. TV-regularization is formulated by using $TV(u)$ and similarity to an original image. It consists of solving the following minimization problem.

$$ u_s = \arg \min_u \left( \int |\nabla v| dx + \frac{1}{2\lambda} \int (v - u)^2 dx \right) $$

where $u$ is the input original image and $u_s$ is the output image.

Since the original image $u$ only contains smooth areas and edges, we refer to the output image $u_s$ as the structure image...
2.2. Bidirectional Similarity

Simalov et al. proposed summarizing visual data using bidirectional similarity [5]. This method resizes the input image, keeping the important areas while preserving its structure.

Let $S$ be an input image and $T$ be a target image. Note that $T$ is always smaller than $S$ because the purpose of the method is summarization. Then, let $N_S$ and $N_T$ denote the number of patches in $S$ and $T$. The following equation formulates the distance between these images.

$$d(S, T) = \frac{1}{N_S} \sum_{P \subset S} \min_{Q \subset T} D(P, Q) + \frac{1}{N_T} \sum_{Q \subset T} \min_{P \subset S} D(Q, P)$$

The first term of Eq. (4) represents completeness, and the second term corresponds to coherence. Completeness measures whether patches of $S$ have been preserved in $T$. Coherence measures how many ‘newborn’ patches in $T$ did not originate from $S$. The result of this method should have a low $d(S, T)$. Therefore, the overall contribution of the value of pixel $q \in T$ to the global bidirectional error measure $d(S, T)$ of Eq. (4) is

$$Err(T(q)) = \frac{1}{N_S} \sum_{j=1}^{n} (S(p_j) - T(q))^2 + \frac{1}{N_T} \sum_{i=1}^{m} (S(p_i) - T(q))^2$$

$n$ and $m$ are the number of patches that are chosen by block matching. The following equation is the update rule of the pixel values for minimization of Eq. (5).

$$T(q) = \frac{1}{N_S} \sum_{j=1}^{n} S(p_j) + \frac{1}{N_T} \sum_{i=1}^{m} S(p_i)$$

By iterating this process and gradually decreasing the target image size, we can obtain the output image as mentioned.

Our method uses this output image for texture image compression. The texture image of the output is restored to the original texture image at the decoder side.

3. PROPOSED METHOD

The outline of our proposed method is shown in Fig. 1. As mentioned above, we obtained the compaction image $Y$, by using the bidirectional similarity method [5], from the input image $X$ as the first step of our proposed encoding algorithm. $X_y$ and $Y_y$ are the luminance images of $X$ and $Y$, respectively. $X_s$ and $Y_s$ can be decomposed into respective structure images ($X_s$ and $Y_s$) and texture images ($X_t$ and $Y_t$) by using TV-regularization [1]. Then, the structure images $X_s$, $Y_s$ and texture image $Y_t$ are compressed by a conventional JPEG encoder with different compression rates. Note that the only discarded component is texture image $X_t$ at the encoder side. Let $I^*$ be the decoded image from $I$; the key feature of our proposed method is that we restore the texture image $X_t'$ by using texture synthesis from $Y_t'$ and the relationship between the two structure images $X_s'$ and $Y_s'$.

3.1. ENCODING

3.1.1. Make Compaction by Image Resizing

As mentioned in Section 2.2, we obtain the small summarization image $Y$ from the input image $X$ by using the bidirectional similarity method [5].

3.1.2. Image Division and Structure Image Encoding

As mentioned in Section 2.1, the structure image $X_s$ is obtained by using TV-regularization from luminance image $X_y$. Therefore, texture image $X_t$ can be obtained by the following equation:

$$X_t = X_y - X_s$$

Since $X_s$ does not contain any texture, it can be highly compressed with the JPEG encoder. In general, the edges of $X_s$ consist of high frequency components and low frequency components. However, we suggest these edges can be restored with a JPEG optimization decoder [2] based on the relationship between the low and high frequency components at the edges. We send $X_s$ which is high compressed to the encoder.
Moreover, we decompose \( Y \) into structure images \( Y_s \) and \( Y_t \). Since texture images only contain high frequency components, a JPEG optimization decoder is not effective for such images. Therefore, we choose a relatively low compression rate for \( Y_t \).

### 3.1.3. Output Bitstream of Proposed Encoder

The output bitstream of our proposed encoder is composed of the following.
- \( X_s \) and \( Y_s \) compressed by JPEG with a high compression ratio
- \( Y_t \) compressed by JPEG with a low compression ratio

Note that there is no explicit mapping information for texture synthesis of \( Y_t \) from \( Y_t \). Our key feature is that, for decoding an original image \( X \), we only need structure image \( X_s \) (which can be highly compressed) and compressed compaction image \( Y = Y_s + Y_t \).

### 3.2. DECODING

The decoding procedure is as follows.
- Structure image \((X_{s'} \) and \( Y_{s'} \)) is decoded by using a JPEG decoder with optimization.
- Texture image \((X_{t'})\) is decoded with texture synthesis from \( Y_t \).
- Luminance image \((X_{y'})\) is decoded by adding \( X_{s'} \) to \( X_{t'} \).

It should be considered that the structure images have block noise and ringing artifacts, because these images are highly compressed by the JPEG encoder. Thus, we use the JPEG decoder with optimization [2] for \( X_{s'} \) and \( Y_{s'} \). The texture image \( X_{t'} \) is restored by texture synthesis (described below) by using the relationship between \( X_s \) and \( Y_s \). Finally, the luminance image is obtained by adding the restored texture image \( X_{t'} \) to the structure image \( X_{s'} \).

Decoded texture image \( X_{t'} \) is synthesized from patches of \( Y_{t'} \). Texture synthesis is based on the map obtained by block matching \( X_{s'} \) and \( Y_{s'} \). Let \( S \) be the source image, \( T \) be the target image, \( w \) be the size of the patch, and \( N \) be the number of pixels (which means \( N = ws \times ws \)). \( \text{map} \) indicates matching from patch \( S \) to \( T \) (s.t. \( P \subset S, Q \subset T \)). Let \( P \) be a patch that contain \( p \) (\( 1 \leq i \leq N \)). We denote this matching as \( \text{map}(S,T)(P,p) \) and formulate it as

\[
\text{map}(S,T)(P,p) = \arg \min_{Q} D(P, Q). \tag{8}
\]

Here, \( D(P, Q) \) is the distance between two patches. In this work, we use the sum of squared differences (SSD) of all corresponding pixels in \( P \) and \( Q \) as \( D(P, Q) \).

Let \( I_{\text{input}} \) be an input image and \( I_{\text{output}} \) be a synthesized image. \( I_{\text{output}} \) is synthesized by using \( \text{map}(S,T) \) and \( I_{\text{output}} \). We define the texture synthesis process as

\[
I_{\text{output}}(p) = \frac{1}{N_p} \sum_{i=0}^{N_p} I_{\text{input}}(\text{map}(S,T)(P_i, p)). \tag{9}
\]

\( N_p \) is the number of patches chosen and contains \( p \). Therefore, the procedure for decoding texture \( X \) is

1. \( Y_{y'} = Y_{s'} + Y_t \)
2. \( X_{t'}(0)(p) = \frac{1}{N_p} \sum_{i=0}^{N_p} Y_t(\text{map}(X_{s'}, Y_{s'})(P_i, p)) \)
3. \( i = 0 \)
4. \( X_{t'}(i) = X_{t'}(i) + X_{s'} \)
5. \( X_{t'}(i+1)(p) = \frac{1}{N_p} \sum_{i=0}^{N_p} Y_t(\text{map}(X_{s'}, Y_{s'})(P_i, p)) \)
6. \( i = i + 1 \)
7. Iterate steps 4-6 while \( i < n \)

Here, \( n \) is the number of iterations. \( \text{map}(X_{s'}, Y_{s'}) \) of step 2 is matching between only structure images. If the pixels of \( X_{s'}(0) \) have the same luminance, they have the same texture. The matching in step 5 is performed between two luminance images (consisting of texture and structure images). By iterating steps 3 through 6, we can obtain a good approximation of \( X_t \) from \( X_{s'}, Y_{s'}, \) and \( X_{t'} \).

Finally, the decoded luminance image \( X_{y'} \) is obtained by

\[
X_{y'} = X_{t'}(n-1) + X_{s'}. \tag{10}
\]

### 4. RESULTS

We tested the efficiency of our proposed method by applying it to test images. In Section 2.1, we described the ROF model [1] for the example of the simplest TV-regularization. However, in this experiment, we used the A2BC regularization to improve the accuracy of the structure/texture decomposition. We set the window size as \( 7 \times 7 \) pixels, which is used for block matching to obtain \( Y \) and \( \text{map} \). To suppress the computational cost for the block matching process, we used PatchMatch [7] to implement our proposed method. \( X_s \) and \( Y_s \) were compressed by JPEG with \( Q = 20 \). \( Y_t \) was compressed by JPEG with \( Q = 75 \). The number of iterations for texture image synthesis \( n \) was 10. For comparison, we also performed conventional JPEG encoding for the original image \( X \). We choose the \( Q \) factor to obtain almost the same data size compared with the results of our proposed method. Table 1 shows the data sizes of the two test images.

<table>
<thead>
<tr>
<th>Image</th>
<th>textures</th>
<th>pond</th>
<th>mandrill</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of ( X \times w ) [pixel]</td>
<td>200 x 200</td>
<td>225 x 300</td>
<td>250 x 250</td>
</tr>
<tr>
<td>Size of ( Y \times h ) [pixel]</td>
<td>79 x 79</td>
<td>92 x 122</td>
<td>128 x 128</td>
</tr>
<tr>
<td>( X_s ) [byte]</td>
<td>1797</td>
<td>2438</td>
<td>4331</td>
</tr>
<tr>
<td>( Y_s ) [byte]</td>
<td>375</td>
<td>487</td>
<td>816</td>
</tr>
<tr>
<td>Total data size of proposed encoder [byte]</td>
<td>2861</td>
<td>3994</td>
<td>7888</td>
</tr>
<tr>
<td>JPEG method [byte]</td>
<td>2882</td>
<td>3994</td>
<td>7912</td>
</tr>
<tr>
<td>Q factor of JPEG method</td>
<td>30</td>
<td>27</td>
<td>30</td>
</tr>
<tr>
<td>PSNR (proposed) [dB]</td>
<td>30.68</td>
<td>28.86</td>
<td>22.69</td>
</tr>
<tr>
<td>PSNR (JPEG) [dB]</td>
<td>32.51</td>
<td>32.01</td>
<td>24.49</td>
</tr>
</tbody>
</table>
4.2. Results and Discussion

The results of the proposed method are shown in Fig. 2. Due to space limitations, we only show the results of image 'textures'. In this experiment, we did not apply any compression for the color components.

(a) Original image
(b) Compaction image
(c) Decoded structure image of proposed method
(d) Decoded texture image of proposed method
(e) Decoded image of proposed method
(f) Decoded image of JPEG method
(g) Expanded image of (e)
(h) Expanded image of (f)

Fig. 2. Experimental results of 'textures'.

Original image of 'textures' and corresponding result of our proposed method shown in (a) and (e). The compaction image is shown in (b) that is obtained from original image. These figures show that our proposed method can reconstruct the texture feature from a compaction image. (c) and (d) is decoded structure image and texture image. Despite the almost-the-same data size, the texture feature of our proposed method has higher accuracy than that of the conventional JPEG encoder (f). (g) and (h) is expanded of small region of our result image and JPEG decoded image. Moreover, the conventional JPEG results exhibit ringing and block noise, while the results of our proposed method do not. However, our method needs much more computational time than JPEG. Implementation of the proposed method was build with Matlab. Entire encoding procedure requires 77.27 second for obtaining the result image in (e). In our current method described in Section 3, compaction image $Y$ is constructed from the viewpoint of two measurements – completeness and coherence. However, we know that the compaction image $Y$ will be used for texture synthesis at the decoder side. Therefore, if we add a third measurement that represents the efficiency for texture synthesis to the compaction image creation (described in Section 2.2) we can expect drastic improvement in the efficiency of our proposed method.

5. SUMMARY AND FUTURE WORK

Images can be decomposed into structure/texture images by using TV-regularization. We propose a method to compress the texture image by using the compaction image of the input image at the encoder side. The luminance image of the input and compaction image are decomposed into structure and texture images. Both the structure images and texture image of the compaction image are sent to the decoder. At the decoder side, the original texture image is reconstructed by using matching between the two structure images. Therefore, the results of our method can reproduce texture features with much greater accuracy than a conventional method such as a JPEG encoder, while having the same data size.

As discussed in Section 4.2, we expect that introducing additional measurement for indicating the efficiency of texture synthesis will lead to drastic improvement in efficiency. The formulation of such a measurement will be our future work.

6. REFERENCES