FINAL REPORT OF DIGITAL IMAGE PROCESSING

Cong-kai Huang, Hong-yu Chen, Yan-cen Lee

ABSTRACT

This paper implements a preprocessing method for DCT-based image compression, independent of the compression process. The method predicts the likelihood of visible distortions at each location. Additionally, we have incorporated Laplacian filtering and applied a boundary tracing algorithm to further enhance the performance of the compressed image, particularly in high-frequency regions.

1. INTRODUCTION

DCT-based compression methods like JPEG and MPEG are widely used in image storage and transmission . However, they often introduce artifacts at low bit-rates. Post-processing at the decoder side can mitigate this, but existing decoders remain unaffected. To enhance image quality at the encoder side, noise reduction before compression is beneficial . We propose a novel filtering method targeting block-DCT-based compression, focusing on reducing ringing noise without significant detail degradation. The approach involves evaluating local characteristics using power spectrum distribution analysis and applying filtering accordingly. This process, independent of the coding process, is compatible with existing encoders. This concise method aims to enhance image quality effectively.At the same time, we also add the Laplacian filter to obtain the boundary to complete the boundary tracking algorithm.

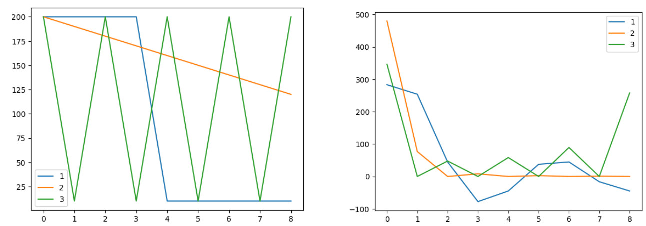
2. NOISE CHARACTERIZATION

According to the findings from the study presented in reference [1], it is observed that visible distortions primarily occur at the edges of objects. Additionally, the image features become distinct after undergoing a DCT transformation. In our research, three signals were employed to simulate various states of the image, along with their corresponding DCT spectra. This can be observed in Figure 1 (Fig. 1).

From the findings in paper [1], we learned that signal two, representing the low-frequency object boundaries, can be estimated using the following formula (1). This corresponds to the region most susceptible to distortion after compression.

$\rho_{mod} = R_{xx}(1)/(R_{xx}(0) + \delta)$ (1)

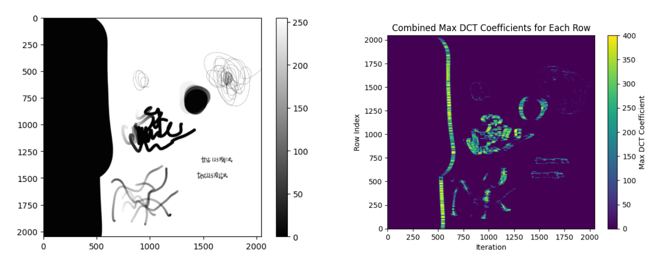
Thanks to XYZ agency for funding.



(a)signal waveform (b) DCT spectrum

Fig. 1. Different signals and their DCT spectra. Signals 1, 2, and 3 represent the image border, image smooth region, and image high-frequency area, respectively.

We also explored an alternative method, which involves finding the maximum value x in the DCT spectrum for frequencies greater than a certain frequency. We observed the magnitude of x, and if it falls within a specific range (a, b), we identify it as the low-frequency transition area in the image. This can be observed in Figure 2 (Fig. 2).



(a)origin image (b) prediction results

Fig. 2. Taking the preceding and succeeding 20 points of each pixel for DCT each time, the maximum value of DCT frequency greater than 32 is extracted. If this value falls between 5 and 10, it is identified as the low-frequency boundary.

3. BOUNDARY TRACING ALGORITHMS

Boundary tracing algorithms, also known as contour tracing algorithms, are used in computer vision and image processing to identify and trace the boundaries or contours of objects within an image. These algorithms help extract the shape or outline of objects, which can be useful for various applications, such as object recognition and image analysis.

Here's a simple explanation of how a boundary tracing algorithm typically works:

- Start Point: The algorithm starts at a specified point on the boundary of an object. This point is often determined using some initial criteria, such as finding the first non-background pixel.
- Traversal: The algorithm traverses along the object boundary, moving from pixel to pixel. The direction of traversal is determined based on the local neighborhood of pixels. Common approaches include following the edge of the object clockwise or counterclockwise.
- Neighbor Pixel Checking: At each step, the algorithm checks the neighboring pixels to decide the next direction to follow. The goal is to stay along the object boundary. Different algorithms may use different rules for determining the next direction, such as following the outer edge or the inner edge of the object.
- Termination: The traversal continues until the algorithm returns to the starting point, completing the boundary trace. Some algorithms may include additional termination criteria, such as reaching a certain number of traced pixels or encountering a specific marker indicating the end of the boundary.
- Output: The result is a sequence of coordinates representing the traced boundary. This information can be further processed or used for various applications.

4. PROPOSED METHOD

Based on the above analysis, we propose the following preprocessing method.

1. A modified auto-correlation coefficient, ρ_{mod}, is computed in the local area near the pixel of interest in the horizontal direction. The area for calculation is determined by the computational amount allowed; usually, 7 to 9 pixels are enough. First, the DC component is subtracted, and then, Rxx (1) and Rxx (0) are calculated.
2. The intensity of the filtering is determined according to ρ_{mod} , so that filtering is applied to places with positive mod. One example is to set the intensity proportional to (ρ_{mod} - ρ_{th}) , where ρ_{th} is a user defined parameter. Then, low pass filtering is applied according to the intensity. In our practice, a signal filtered by a linear low pass filter was mixed with the original signal with an intensity proportional to (ρ_{mod} - ρ_{th}).
3. Repeat step 1 and 2 for each pixels in the image.
4. Do step 1 through 3 again for vertical direction
5. Utilize the boundary tracing algorithm to extract the edges of the image. Then, overlay these edges onto the modified experimental image to generate image overlay1.
6. Overlay image overlay1 onto the image that has undergone Laplacian filtering to obtain image overlay2.

If the image is in color, the above process is applied to each color plane.

In the case of color images with different sampling ratios for each color components, such as YUV-4:2:0, this process should be applied to each color plane according to each resolution.

5. EXPERIMENTAL RESULT

In this section, the effect of the proposed method is briefly reviewed. Fig. 3 is original image in Fig. 5 after boundary splitting and the image after laplacian filtering of 200*200 pixels Fig. 4 is original image in Fig. 5 after boundary splitting and the image after laplacian filtering of 500*500 pixels Fig. 5 is original image and image after ideal lowpass filtering of 500*500 pixels Fig. 6 is original image after filtering with the proposed method and the edge image in Fig. 4 overlay with modified LPF image of 500*500 pixels Fig. 7 is overlay image1 500 overlay with laplacian image in Fig. 4 Fig. 8 is original image and image after ideal lowpass filtering of 200*200 pixels Fig. 9 is original image after filtering with the proposed method and the edge image in Fig. 3 overlay with modified LPF image of 200*200 pixels Fig. 10 is overlay image1 500 overlay with laplacian image in Fig. 3 Fig. 11 is comparison of different methond on its PSNR of 200*200/500*500 pixels

Fig. 3. edge200 and edge 500

Fig. 4. edge200 and edge 500

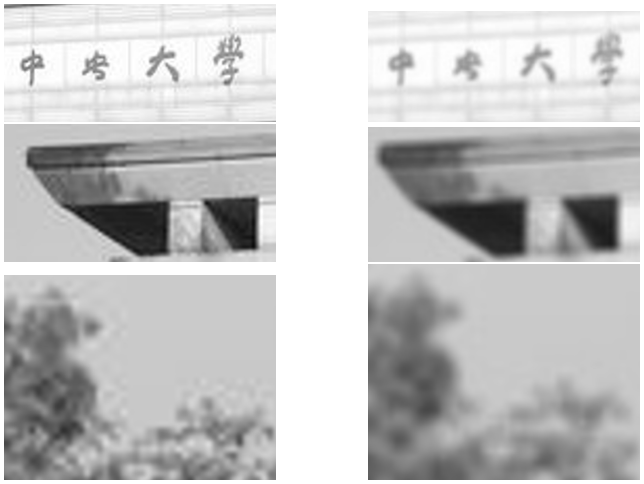


Fig. 5. orignial image and 500 ideal LPF 500

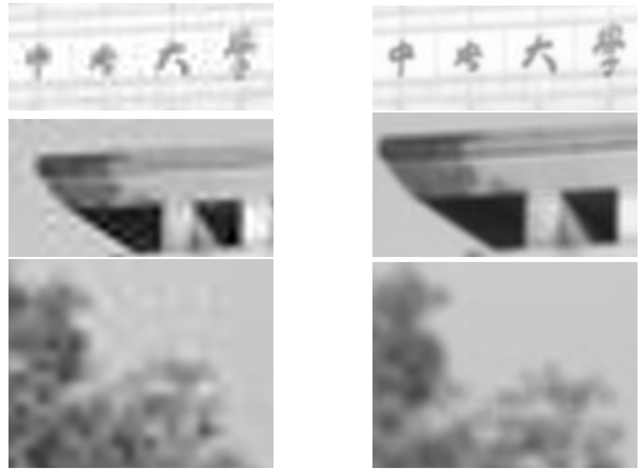


Fig. 8. orignial image and 200 ideal LPF 200

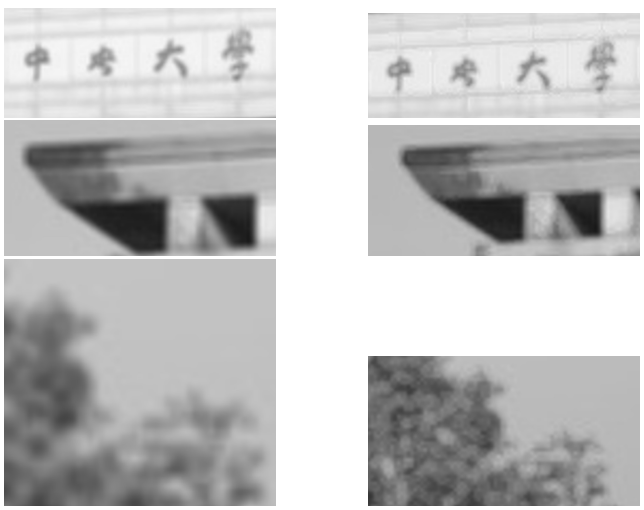


Fig. 6. modify LPF 500 and overlay image1 500

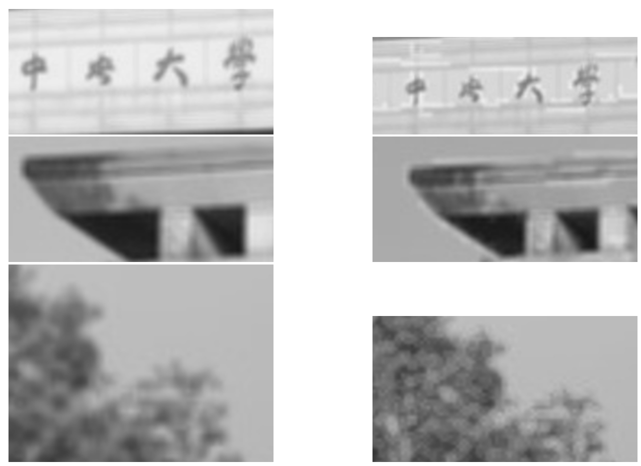


Fig. 9. modify LPF 200 and overlay image1 200

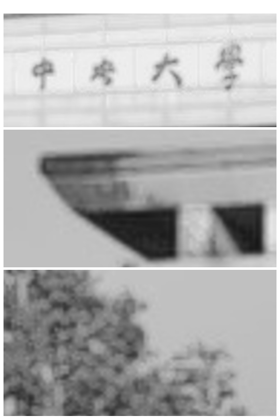


Fig. 7. overlay image2 500

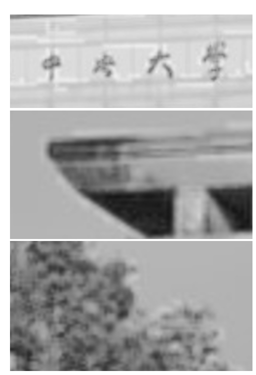


Fig. 10. overlay image2 200

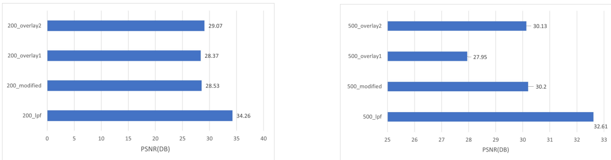


Fig. 11. PSNR 200 and PSNR 500

6. CONCLUSION

DCT-based compression methods, such as those employed in JPEG and MPEG, are extensively utilized for image storage and transmission. However, they often introduce artifacts, especially at low bit-rates. Although post-processing at the decoder side can alleviate this issue, it leaves existing decoders unaffected. To improve image quality at the encoder side, implementing noise reduction before compression proves beneficial. We propose a novel filtering method designed for block-DCT-based compression, with a specific focus on reducing ringing noise without causing significant detail degradation. This method involves assessing local characteristics through power spectrum distribution analysis and applying filtering accordingly. Importantly, this process, independent of the coding process, is compatible with existing encoders. Our concise method aims to effectively enhance image quality.

In this report, we have successfully implemented the majority of methods discussed in the literature, including the boundary tracing algorithm. However, it is important to note that the utilization of the methods outlined in the paper still requires a significant amount of computation time. Therefore, our future objective is to enhance the computational efficiency of our program, enabling the efficient processing of a large number of images.

[1]

7. REFERENCES

[1] Munenori Oizumi, "Preprocessing method for dct-based image-compression," in *IEEE Transactions on Comsumer Electronics*. IEEE, 2006, vol. 52, pp. 1021–1026.