CHIBA INSTITUTE OF TECHNOLOGY
GRADUATE SCHOOL OF INFORMATION AND COMPUTER SCIENCE
千葉工業大学大学院
情報科学研究科 情報科学専攻

DOCTORAL THESIS
博士学位論文

A Study on Sparse Representation Dictionary Design Method using Multi-class K-SVD and Its Application to Image Coding

マルチクラス K-SVD によるスパース表現辞書設計と画像符号化への応用に関する研究

Student ID 1489501
Name WANG Ji
氏名 王 冀

Supervisor Prof. YASHIMA Yoshiyuki
指導教員 八島 由幸 教授

March, 2021

Summary of Doctoral Thesis

Course	Student No.	SURNAME, First name
Information and Computer Science	1489501	WANG Ji

Title:

A Study on Sparse Representation Dictionary Design Method using Multi-class K-SVD and Its Application to Image Coding

Keywords:

Image Coding, Sparse Coding, K-SVD, OMP, Multi-class

Summary:

In recent years, due to the higher resolution of images and the spread of SNS, the ratio of image data to total network traffic has increased remarkably. Therefore, the development of high-efficiency image coding scheme that exceeds the conventional international standards is expected. Recently, the next-generation international standard VVC (Versatile Video Coding) is being studied, but VVC uses the same framework as the conventional international standards that combine the intraframe/interframe prediction and the transformation based on DCT. However, the Discrete Cosine Transform (DCT), which is used as a spatial correlation removal technology in JPEG, H.265/HEVC and VVC, is not always the optimal dictionary, i.e. set of atoms, for a specific image to be encoded. In this paper, we propose a new learning-type image coding scheme, in which instead of a fixed dictionary based on the DCT, the dictionaries adaptively trained for various images by sparse coding are utilized. Sparse coding finds the atoms that compose the dictionary and the weighting coefficient for each of those atoms so that the squared error when expressing the original signal as a weighted linear sum of the atoms is minimized while satisfying the constraint condition that the number of non-zero weighting coefficients is less than a certain number. In the sparse coding, a high compression rate can be achieved by setting the number of non-zero coefficients to be very small. In this paper, we propose three technologies when applying sparse coding to image compression, an adaptation method that reflects the local features of an image, a quantization method for the sparse coefficients, and an entropy coding method for the sparse coefficients obtained by sparse coding.

First, we propose a novel multi-class dictionary design method. The small blocks (that is, learning vectors), which compose the images, are divided into multiple classes according to their features, and a dictionary is designed for each class. In the proposed method, the

optimum dictionary is designed by repeating the class update stage of all learning vectors and the K-Singular Value Decomposition (K-SVD) based dictionary update stage of each class. For image coding, the learning vectors are obtained by dividing training images into small blocks. When encoding/ decoding, the sparse coefficients are obtained by using an optimum dictionary from the pre-shared multiple dictionaries in the encoder and decoder for each small block in the target image to be encoded, and they are sent to the decoder after quantization and code assign. It is clarified that the proposed method can reduce the bit rate by up to 48 % compared to the conventional fixed classification method. Then, we show that the probability distribution of quantized sparse coefficients depends on the corresponding atom features and the number of sparse coefficients in that block, and the amount of occurred bits can be reduced by 6 % by introducing the efficient context-adaptive entropy coding method. Finally, we also propose a quantization matrix design method that determines the quantization width of the sparse coefficient based on the relationship between the complexity of the DCT atoms and the complexity of the atoms designed by K-SVD. The proposed method can perform adaptive quantization of a sparse coefficient according to the characteristics of its corresponding atom. Therefore, the perceptual image quality has improved by about 0.3 points in most ranges of 5-stage MOS (Mean Opinion Score) compared to the conventional uniform quantization.

The three elemental technologies established in this paper, that is, the multi-class dictionary design method in sparse coding, the adaptive code assignment method for sparse coefficients and the sparse coefficient quantization matrix design method, open the way for the practicality of image compression with sparse coding. This is expected to contribute significantly to future research and development as a new direction for learning-based image coding schemes beyond many conventional frameworks.

Graduate School of Information and Computer Science, Chiba Institute of Technology

博士論文要旨

専攻	学籍番号	氏名
情報科学	1489501	王　冀

論文題目

マルチクラス K-SVD によるスパース表現辞書設計と
画像符号化への応用に関する研究

キーワード

画像符号化，スパースコーディング，K-SVD，OMP，マルチクラス

論文要旨

画像の持つ情報量は極めて膨大であり，効率的なネットワーク伝送や，デバイスへの蓄積のためには，圧縮技術が必要不可欠である．現在世の中では，画像圧縮国際標準として，静止画に対する JPEG や，動画に対する H.265/HEVC が，放送・通信・家電等の分野で広く用いられている．しかしながら，近年の，画像の高精細化や SNS の普及により，画像情報がネットワークトラフィックに占める割合は年々著しく増加しており，さらなる高効率圧縮技術の開発が期待されている．

JPEG や H.265/HEVC においては，空間的な相関除去方法として，離散コサイン変換（DCT）に基づく手法が継続的に採用されている．しかしながら，ある特定の画像を符号化対象として考えると，DCT は必ずしも最適な辞書（すなわち，基底の集合）とはならない．本論文では，スパースコーディングを用いて様々な画像に対して辞書学習を行い，得られた辞書を用いて符号化を行うことで，DCT よりも優れた符号化効率の達成を狙っている．スパースコーディングとは，原信号を，基底の重み付き線形和で表現する際に，非ゼロ係数の個数をある一定の数以下にした時に，原信号と復元信号の二乗誤差が最小になるように，基底及び重み係数を求めるものであり，非ゼロ係数を極めて少数に設定することで大幅な情報圧縮を図ることが期待できる．スパースコーディングを画像圧縮に適用する場合，①画像の局所領域の特徴にどのように適応させるか，②スパースに分布した非ゼロ係数にどのように符号を割り当てて情報量削減するか，③復号画像の主観画質を向上するためにはスパース係数をどのように量子化すればよいか，を明らかにする必要がある．本論文ではこれらの 3 つの観点にフォーカスを当てて検討を行い，課題を解決す

る新手法の提案とともに，実験により提案手法の有効性を確認している．

まず第 1 は，画像符号化に適したスパース表現可能な辞書設計手法に関する取り組みである．入力データのスパース表現を可能にする辞書を学習するための手法として K-SVD（K-Singular Value Decomposition）を用いる．K-SVD によって設計された辞書の画像表現能力は、学習に用いる画像データの特徴に大きく依存する．従って，入力データを複数のクラスに分類し，クラスごとに辞書を学習する手法を採用する．しかしながら，従来検討されているマルチクラス辞書設計方法は，事前に定められた特徴量を用いた分類に固定されており、画像データの分類段階と辞書学習段階の関係が考慮されていない．このため，本論文では，K-SVD による全学習ベクトルのクラス更新段階と各クラスの辞書更新段階を繰り返し処理によって最適化する新しいマルチクラス辞書設計法を提案する．さらに，設計された辞書を用いた画像符号化システムとして，学習で得られた複数の辞書をエンコーダとデコーダで共有し，画像中の小ブロックごとに最適なクラスの辞書を選択してスパース表現符号化する構成法を提案する．実験の結果，提案手法は，従来の固定的なクラス分けに比較して，BD-rate で最大 48%，BD-PSNR で最大 1.6 dB の符号化効率向上を達成できることが明らかとなった．

第 2 の取り組みは，スパースに分布する非ゼロ係数のエントロピー符号化に関する提案である．本論文では，理論的および実験的観点から重み係数の統計的特性を詳細に分析し，その分析に基づいてスパース係数の効率的なエントロピー符号化方法を提案する．本検討では，非ゼロ係数のレベル値と，非ゼロ係数間のゼロラン長に着目する．詳細な解析により，非ゼロ係数レベル値の分布特性が，ブロック内の非ゼロ係数の個数およびその非ゼロ係数に対応する基底の特徴によって異なることを示し、非ゼロ係数レベル値に対しては，これらに基づくコンテキスト適応符号化が有効であることを明らかにする．さらに，ゼロラン長は，基底の全変動特徴に基づいて基底を並び替えることによって発生確率に偏りを持たせることができ，効率的な符号化が可能であることを示す．実験の結果，提案手法は従来手法と比較して，約 6%の発生ビット量削減を達成できることを明らかにした．

第 3 の取り組みは，スパース係数の新しい量子化手法に関するものである．スパース表現符号化では，スパース係数を量子化して伝送する．一般的に，JPEG や H.265/HEVC などで用いられる DCT 係数に対しては，高周波基底に対する係数の量子化幅を，低周波基底に対する係数の量子化幅よりも大きく設定することで，視覚的な画質を向上できる．一方で，ス

パース表現を可能とする辞書は，それに含まれる各々の基底が複雑な周波数成分を持ち，DCT 基底や DFT 基底のように規則的な配列を構成していない．そのため，従来の検討では，どの基底に対する係数も同じ量子化幅で量子化する手法が用いられてきた．本論文では，K-SVD によって設計された辞書の各基底の複雑度を定義し，HEVC の DCT 基底の複雑度との類似度に基づいてスパース係数の量子化幅を決定する量子化マトリクスを設計する．この方法により、基底の特徴に応じて適応的に係数量子化を実行できる．主観評価実験により，従来の一様量子化に比較して，同じ圧縮率における 5 段階 MOS（Mean Opinion Score）が約 0.3 ポイント向上し，視覚的画質が改善されることが示された．

以上を総合して，本論文では，スパースコーディングを応用した学習型画像符号化という画像圧縮への新しいアプローチの提案と，それを実現するための 3 つの必須要素技術，すなわち，辞書設計・係数量子化・符号割り当てについての具体的な方法を明らかにし，従来方式を上回る符号化効率を達成可能であるという知見を得ることができた．

千葉工業大学大学院情報科学研究科

Acknowledgements

I would like to extend my deep gratitude to all those who have offered me practical, cordial and selfless support in writing this thesis.

Firstly, I am extremely grateful to my supervisor, Prof. Yashima Yoshiyuki. He guides me, influences me and helps me in the process of writing this thesis. It is with his patience, generosity, and encouragement that I finally write, revise and finish my thesis. I also would like to thank Professor Tadahiko Kumamoto, Professor Junichi Imai, Associate Professor Yusuke Manabe of Chiba Institute of Technology, and Professor Kazuto Kamikura of Tokyo Polytechnic University for their valuable opinions to complete my thesis.

Secondly, I am much obliged to all teachers who ever taught me during the years of my postgraduate studies. It is these distinguished professors who give me the opportunity to walk into the academic world. To study with them will always be an honor for me and their instructions will follow me in my future studies.

Thirdly, I would like to thank the staff in Student Affairs and Educational Affairs who made my years in a foreign country feel like home as well. And my dear friend, Gong Chao, who is strong-willed and always passionate about life, has also offered me great help. He taught me how to balance the stresses of research and life during the years we studied for our degrees together.

Lastly, I also want to thank my family, my parents, who have shown me the way of life, my lovely daughter Nian-nian, who gives purpose to this way, and my wife Xiang-wei, who constantly paves this way with love and kindness. It is their understanding and patience that make this thesis possible.

Contents

List of Figures

List of Tables

List of Abbreviations

ACR Absolute Category Rating

AVC H.264/Advanced Video Coding

BD metric Bjøntegaard Delta Metric Model

CABAC Context Adaptive Binary Arithmetic Coder

CB Coding Block

CGM Consumer Generated Media

CNN Convolutional neural network

CTB Coding Tree Block

CTU Coding Tree Unit

CU Coding Unit

CU Coding Unit

DCT Discrete Cosine Transform

DPCM Differential Pulse Code Modulation

DSIFT Dense Scale-Invariant Feature Transform

DST Discrete Sine Transform

DWT Discrete Wavelet Transform

EOB End of Block

EVD Eigenvalue Decomposition

FPS Frames Per Second

GPU Graphic Processing Unit

HEVC H.265/High Efficiency Video Coding

HVS Human Visual System

IoT Internet of Things

JPEG Joint Picture Experts Group

JVET Joint Video Experts Team

K-SVD K-Singular Value Decomposition

KLT Karhunen-Loeve transform

ML Maximum Likelihood

MOS Mean Opinion Score

MOS Mean Opinion Score

MPEG Moving Picture Experts Group

MSE Mean Squared Error

MTF Modulation Transfer Function

MV Motion Vector

OMP Orthogonal Matching Pursuit

PB Prediction Block

PCA Principal Component Analysis

PSNR Peak Signal to Noise Ratio

PSNR Peak Signal-Noise Ratio

PU Prediction Unit

RD curve Rate-Distortion Curve

RLE Run Length Encoding

SHVC Scalable Extension of HEVC

SIFT Scale-Invariant Feature Transform

SSIM Structural Similarity Index

SSIM Structure-Similarity Index

TB Transformation Block

TU Transformation Unit

UHD Ultra-High Definition

VCEG Video Coding Experts Group

VCEG Video Coding Experts Group

VLC Variable Length Coding

VVC H.266/Versatile Video Coding

Introduction 1

Contents

1.1 Background of the Study

1.1.1 Environment surrounding image communication

Image communication systems transmit and receive image information over a network, in which not only voice but also information such as images, videos, texts, and charts are converted into electrical signals via image/video equipment, and they are visualized on the receiving side and perceived by the human eye. Therefore, we can regard image communication as communication via visual information.

The human visual system is the main channel for us to obtain various information, and the need for visual information in communication is becoming more and more important. Due to recent advances in communication technology and related services, our demands have changed from voice-only communication to communication that includes both image and voice. Image communication, which integrates voice, data, and images into one communication service, has become a hotspot in the communication field and is becoming more and more widely used in various industrial fields such as video conferencing, remote medical systems, and online education.

The development of image communication is closely related to the following three factors:

1. Efficient image compression technology,
2. Reliable image communication infrastructure technology,

3. Secure image communication technology.

I would like to clarify the first factor in terms of the amount of image information. In relation to the second factor, I will give an overview of the development of network technologies and trends in image communication services and image processing devices. The third factor includes important technologies such as image encryption and digital watermarking, but it is out of the focus of this paper, so the recent trends of the first and second factors are introduced below.

1.1.2 Amount of image information

The amount of information in an image depends mainly on the resolution of the image, the color expression capability of each pixel and the number of frames per second (FPS) of video to be transmitted.

With the new generation of photography and video equipment, the mainstream photo attributes have evolved from Full HD (2K) to 4K, and even to 8K. 2K, 4K and 8K means the number of horizontal pixels of an image. The standard pixel size for Full HD is 1920×1080, which is about $2K \times 1K$. The 4K resolution is doubled horizontally and vertically on top of the 2K image, which is equivalent to 4 times the number of pixels in a 2K resolution image. 8K increases the sampling density on top of 4K, which is equivalent to four 4K images or sixteen 2K images stitched together.

In the second half of 2012, the International Telecommunication Union (ITU-R) issued the BT.2020 standard[2] for the new generation of Ultra-High Definition (UHD) video production and display systems. BT.2020 standard significantly improved the performance of video signal specifications, compared to the previous generation of BT.709 standard, and have promoted the further spread of 4K UHD home display equipment in the field of television broadcasting and consumer electronics. For example, the color depth has been increased from 8 bit in BT.709 to 10 bit or 12 bit, where 10 bit is for 4K systems and 12 bit is for 8K systems. This enhancement plays a key role in enhancing the overall image in terms of color levels and transitions. The color gamut is also much larger than the BT.709 standard, enabling richer colors to be displayed.

The BT.2020 standard not only specifies the UHD display resolution of 3840×2160 and 7680×4320 with a 16:9 display ratio, but also extends the supported FPS limits to 120p, 60p, 59.94p, 50p, 30p, 29.97p, 25p, 24p, 23.976p, where p is the initial letter of "progressive scanning". The interlaced scanning has finally been eliminated in BT.2020, and all images in UHD are based on progressive scanning, which is undoubtedly a historic breakthrough

and further enhances the fineness and smoothness of UHD images.

The constant improvement in the "quality" of an image causes an exponential increase in the amount of information generated by the image. For example, the amount of information of uncompressed HD video is about 1 Gbit/s, while that of 4K/30P and 8K/60P videos becomes 4 Gbit/s and 32Gbit/s respectively, which is extremely large.

1.1.3 Progress in network technology

Another trigger for the image information explosion is the acceleration of recent network technology upgrades. At MWC in 2018, 5G has become a hot topic all over the world, and domestic interest in 5G has become very high. 5G is abbreviated as "5th generation mobile communication network". Since mobile communications have long been an integral part of every aspect of our daily lives, people's expectations for the arrival of 5G are enormous. We can check the news almost every day, check emails on the smartphones, and even watch sports games from around the world using iPad. These experience in daily life is actually backed by a complete and powerful mobile network communication technology. Without mobile network communication protocols and technologies, today's versatile entertainment, learning and work life is impossible.

With mobile network updates, the most obvious change is the increase in bandwidth, the benefits most directly felt by the average user.

Nearly two decades ago, there was no feature-rich "smart phone" in the hands of the people – the more common name at the time was "mobile phone/cell phone" or "handphone" to distinguish it from a landline. At that time, "mobile phones" were really "simple communication tools", so they could hardly be used for anything other than sending and receiving calls and text messages.

At that time, the demand for mobile communication networks was limited to text messaging (text transmission) and telephone (voice communication), so even a 2G network with a bandwidth of only 150kbps had sufficient capacity. But for larger data files such as color pictures, 2G networks were overwhelmed.

The usage pattern of mobile phones had changed with the times, and various technologies have been installed in mobile phones. A camera was installed on the back of the mobile phone so that it can be used as a digital camera. With the development of color screens, various types of simple mini-games on mobile phones have been enthusiastically sought after and welcomed. For these applications, 2G networks are "insufficient" in the face of increasing demand for mobile phones (sending and receiving images, downloading games, etc.).

Therefore, a 3G network was developed. 3G networks have a bandwidth of 1 to 6 Mbps, and the transmission speed is dramatically faster than that of 2G networks. 3G network makes it possible to transmit color images and download games within the acceptable time.

Soon, however, there was a new demand to mobile phone. With the introduction of smart devices, users wanted to watch videos, live streams and play games with better picture quality on their smart mobile devices, and 3G networks had been not enough anymore for these purposes. Then, a 4G network with a bandwidth of 10-100Mbps was developed. Today, 4G networks has been widespread in many countries, and enables HD quality image transmission and video distribution.

The 5G mobile network communication, which will become widespread in the future, will enable us to respond to new user demands and business environments. On the other hand, artificial intelligence algorithms are advancing day by day, and as cloud computing becomes a mature technology, it is required to transmit a huge amount of data used for machine learning. Machine learning, especially deep learning, has demonstrate excellent performance in the fields of image recognition and image analysis and it has been already familiar to us. In order to build a better machine learning system, it is necessary to acquire a large amount of learning images and transmit them to a cloud computer. On the other hand, the new era of 4K and 8K image quality poses new challenges for data transmission over networks. The 4K moving image data requires 30-40Mbps even after compression, and 8K moving image data requires 80-100Mbps[3].In the 5G era, it is predicted that ordinary users will be able to use mobile networks to watch higher resolution videos, movies, and smoother videophones. However, at the same time, the amount of data transmitted over the network is expected to increase further, requiring new technologies that enable more economical transmission than existing image compression technologies.

1.1.4 Trends in services and devices

In 2009, the digital TV signal in the United States completely replaced the analog TV signal; in 2012, Japan became the first Asian country to completely switch to digital TV signal; Europe and Australia have subsequently completed the upgrade from analog to digital TV signal. People can enjoy the visual enjoyment of full HD images without even leaving their homes. Traditional video sharing sites, such as YouTube, have accelerated their video "picture quality" upgrade, with 4K video playback starting in 2013 and 8K options added to PC players starting in 2015.

The upgrade of service also drives the upgrade of video image related equipment, when the concept of 4K, 8K permeates into thousands of households, the next "growth point"

of video service is handed over to the innovation and development of the hardware field. Panorama cameras, drones, light field cameras and other hardware have made a breakthrough in terms of viewing angle, shooting position, imaging method, breaking the existing image and video in the public mind, bringing us a different visual experience.

1.2 The Need for Image Coding

1.2.1 The necessity of image coding

With the development of modern communication technology, the type and amount of image information and data required to be transmitted has increased dramatically. It would be difficult to disseminate and apply this image information without compression of them.

On the basis of the above background, the recent image environment is undergoing "qualitative enhancement" and "quantitative enhancement".

"Qualitative enhancement" is evidenced by three aspects. First one is the spread of 4K and 8K high-resolution images and video contents, higher gradation of luminance and color representation such as HDR is the second aspect. The third one is that multiple viewpoints contents have become to be available such as 360° panoramic video, drone video, and multi-view video. The emergence of multiple video formats to match them is as well a part of this enhancement.

As for the "quantitative enhancement", the advent of various simple and highly functional image/video applications has made it possible for ordinary users to upload the images filmed or created by themselves (CGM: Consumer Generated Media) to the Internet and circulate via social networking services in large quantities. The amount of video data on the Internet is growing rapidly. Cisco Visual Networking Index 2018[4] estimates that video data will account for 82% of internet traffic by 2022.

1.2.2 The directions of image coding

Typical image coding process is based on the flow of block division, prediction, transformations, transformation coefficients quantization, and code-assignment. This framework has had a great influence on the subsequent international standardization of video coding, and has been followed up to H.265/HEVC(HEVC). Although the current terrestrial digital broadcasting uses MPEG-2, the next generation broadcasts, mainly 4K and 8K, use HEVC with its high compression ratio. In Internet video streaming and consumer electrical de-

vices such as digital video cameras and Blu-ray recorders, H.264/AVC (AVC) and HEVC are becoming more and more popular.

1.2.2.1 From how to rate "good images"

Television broadcasts and storage media use lossy compression coding, such as JPEG, MPEG-2 and HEVC. In other words, the goal is to achieve "zero perceptual loss", that is, each pixel value of the decoded image is not completely coincident with the corresponding pixel value of the original image, but there is no visual degradation from the original.

The effectiveness of these image compression methods is evaluated by how closely they approximate the original image, the medical and artistic fields have the most stringent requirements for this kind of metric, often requiring the decoded image to perfectly match the original image. Therefore, they mainly use lossless compression, in which "original image fidelity" is utilized as the evaluation metric.

However, given the limitations that can be encountered in practice in the communications field, for example due to network performance and storage device capacity, any lossless compression is not sufficient. It is important to perform RD optimization to minimize the degradation under a defined compression ratio. The "R" of RD optimization is the rate, or the amount of code generated, and the "D" is the distortion of the decoded image, but the question is what measure of "D" should be used here. In the case of "original image fidelity" as pixel values, the distortion can be quantified by the PSNR, which is defined by the difference between the decoded image and the original image. On the other hand, when the visual perception is taken into account, the Structural Similarity Index (SSIM)[5], etc., that are compatible with human vision is more appropriate.

This research direction of optimizing visual perception for the viewer requires some expansion on existing image coding standards. For example, on the encoding side, using filters such as the non-local means filter for noise reduction while maintaining edge information, "Coefficient quantification matrix" and "Domain Adaptive Quantization" developed for the "block artifact" are the typical processes. While on the decoding side, deblocking filters to remove coding noise and super-resolution processing to improve the sharpness of blurred images are some examples. Additionally, visual optimization improves the subjective evaluation value, but may reduce the PSNR and SSIM.

In addition to "original image fidelity" and "visual perception fidelity", another new research direction "visual impression fidelity" is to pursue a kind of sensory "unity" between the decoded image and the original image. In other words, although the data and the appearance are different from the original image, it is still good if there is no discomfort.

There are three main categories of coding methods based on "visual impression fidelity": image-generating coding, flexible display-oriented coding, and entity-mining coding methods.

Image-generating coding

The idea of image-generating coding is to significantly reduce the information on the encoding side and "generate" the reduced information by some methods on the decoding side. Obviously, this kind of processing requires high computing power on both the encoding and decoding sides, and it is difficult to make it commonplace in a short time. This kind of research is more inclined to face the future communication environment such as more abundant big data resources to provide reference images, faster network communication environment to ensure high-speed search and download of resources, and more computing power of terminal communication equipment to provide the sufficient decoding.

Based on the above idea, the emphasis is on how to select information suitable for "scaling down" at the encoding side.

One idea is to cut the color information drastically on the encoding side, and then reproduce the image color on the decoding side with a few remaining"hints". This applies to a technique called "Colorization", which is a category of techniques that includes methods for coloring an image by diffusing color using a small quantity of color-marked information[6, 7]. Another method is to "import" color information by searching the database for pixel patterns with similar characteristics using grayscale components[8]. There are also methods for color inference through deep learning of large-scale pairing of monochrome and color images[9]. Compared to the latter two, the former requires some additional color cueing information in the transmitted information, but also ensures that the coloring of the decoded image and the original image remains uniform. The latter two can achieve higher image compression efficiency in terms of reducing amount of bits to be transmitted since only information from the grayscale image is needed, but there is a risk of generating a completely different coloring style image from the original one, making it difficult to apply to practical use.

Another kind of information suitable for scaling down at the encoding side is the spatial information, known as resolution, of the image itself. From an informatics point of view, image down sampling is approximately equivalent to a low-pass filtering of the image. For example, the use of wavelet transform down sampling is, in essence, to delete the high-frequency information in the original image, and the remaining low-frequency information forms a smaller image. The downscaled images are very efficient for compression. The orig-

inal image is downscaled on the encoding side, and the downscaled image is encoded and transmitted over the network. The lower resolution of the image to be encoded makes it suitable for even narrower bandwidth networks. On the decoding side, the reduced image is decoded and reconstructed to the resolution of the original image using super-resolution technology. This method enables high-resolution image transmission with a low bit rate. In fact, some of the existing 4K and 8K image transmission is made use of this kind of technique[10]. Incidentally, some 4K TVs out there use similar techniques to enlarge the image when displaying 2K sources.

Similar to the above idea of using super-resolution images for coding is texture synthesis coding. Texture synthesis coding focuses the efficient representation for the large region consist of complex textures contained in an image to be coded. The texture region is represented by repeating the basic pattern, even if it is composed of a complex combination of various frequency components. Taking advantage of this property, the encoder extracts a smaller partial region, which is called a "patch" from a large texture region, encodes the patch, and transmits to the decoder. In the decoder, the large texture area is reconstructed by performing texture synthesis using the decoded patch[11]. Since the human eye is not sensitive to changes in texture details, this technique can produce an image which gives similar visual impression with the original image. However, the method of selecting an appropriate patch has not been sufficiently researched, and there are still problems in putting the texture coding into practical use.

Flexible display-oriented coding

With the spread of Internet of Things (IoT) and 5G devices and the advancement of flexible display material technology, the viewing environments of video are diversifying, and the need for image coding technology that can flexibly respond to different viewing environments is increasing.

The Scalable Extension of HEVC (SHVC)[12] allows us to decode images at different bitrates, frame rates, and resolutions from a single compressed bitstream, therefore SHVC can flexibly adapt to many viewing environments with various network speeds and various display resolutions.

However, SHVC is not sufficient for adapting to displays with various screen aspect ratios and for non-rectangular displays. One approach to solve this problem is a content scalable video coding. In other words, only non-important objects are compressed spatially and temporally without any visual impression degradation, while important objects in the image are preserved. Some techniques such as seam carving[13] are applied to implement

the content scalability.

1.2.2.2 From employing AI and machine learning

In addition to the expansion of research directions brought about by the change of image evaluation standards, the rapid progress of artificial intelligence and machine learning-related technologies, such as deep learning, has also impacted image coding technology in recent years. Some attempts to combine it with existing standards are also very common.

The encoding side of traditional image compression mainly consists of four steps: prediction, transformation, quantization, and encoding, which contain a large number of variables that need to be controlled in the actual application process, and how to decide which variables to use for different encoding targets has become a difficulty. To achieve specific algorithm tuning by manual effort can almost be seen as an impossible task, so using computers to automatically perform algorithm optimization and fine-tuning is a worthwhile direction to try.

Convolutional neural network (CNN) technology enables the analysis and pre-diction of the target image by constructing deep neural networks through multi-layer stacking of filters with several different features. As its reliability has been widely proven in recent years, several attempts have been made to combine it with image coding techniques, the most representative of which is the application to "division mode decision" and "intra-frame prediction mode decision".

In the image coding standard, for the target image, the image should first be divided into different-sized blocks according to its local features. This division is not purely based on the statistical features of the pixels in the image, but should take into account the amount of entropy generated by different division modes, and select the mode with the lowest entropy on the premise of guaranteeing the decoded image quality. This undoubtedly requires a lot of computation time. Reference [14] proposed a method to determine the division mode using CNN for this problem. By using a large number of data sets such as entropy generated by Coding Unit (CU) in different modes, and by pre-training the CNN, the neural network has the ability to infer the entropy generated by different division modes from the image features and then determine the optimal division mode, which achieves more than 50% acceleration without degrading the performance of the whole coding architecture.

The conventional prediction method employed as the international standard is achieved by adaptively switching several prediction patterns that are predefined in advance. Taking the intra-prediction as an example, AVC has 9 prediction patterns and HEVC has 35 prediction patterns for a block to be predicted, of which the prediction pattern that best approx-

imates the target block is used. However, the accuracy of approximation is limited because the prediction patterns are predetermined. Using CNN, there is a possibility of generating accurate predictions that cannot be achieved by such predefined patterns[15] . More concretely, it is enough to prepare a large number of pairs of predicted blocks (true values) and their encoded neighbor pixels, and to learn CNN parameters so that neighbor pixels are used as CNN inputs and the CNN output approximates the true value as much as possible.

This idea can be extended to inter-frame predictions as well. Jimbo et al.[16] used CNN to learn a transformation matrix for estimating the target block by deforming the blocks in the frames before and after the target block. This transformation matrix can achieve not only conventional motion compensation, but also prediction that compensates for translation, scaling, and blur at arbitrary accuracy, and achieves better prediction performance than HEVC.

Discrete Cosine Transform (DCT) is used for linear transformation of image in the existing international standards, because the processing framework using DCT atoms is considered to be more "stable" in terms of good transformation efficiency for different kinds of images. In other words, although the DCT atoms has achieved a good balance between generality and efficient signal expression from a statistical point of view, it is not necessarily the optimal solution for a specific image coding target. Bryt and Elad have discussed this point of optimization[17]. They have proposed a more efficient image compression method using K-Singular Value Decomposition (K-SVD)[18], which uses sparse coding and Orthogonal Matching Pursuit (OMP)[19] to develop dedicated optimized atoms for a specific type of image. This approach offers the possibility to go beyond the compression efficiency of the DCT atoms.

In addition to the optimization explorations on the above-mentioned two steps of prediction and transformation, Takamura has conducted a study[20] on the use of the Genetic Algorithm for the integrated optimization of orthogonal transformation, quantization, and intra-loop filter. The goal of his study is to automate the variables in the whole process of traditional compression coding. It is demonstrated that automatic optimization of such large-scale variables is possible with high performance computing power and parallel processing, such as Graphic Processing Unit (GPU) technologies.

1.3 Position and Purpose of This Study

1.3.1 Motivation

Image coding technology is one of the key technologies for communication services, broadcasting and many storage devices. JPEG and H.264/AVC are widely used for many currently provided image and video services. H.265/HEVC, the latest international standard, is also spreading as a coding scheme for ultra-high resolution video. The transformation process for expressing an image to be encoded as a weighted linear sum of the atoms is one of the key technology elements for image compression. Most of the conventional image coding standards adopt transforms based on DCT. This is because DCT gives a good approximation of the Karhunen-Loeve transform (KLT) under the condition that there is high correlation among neighboring pixels, which is a known statistical property of many natural images and videos. However, DCT is not efficient enough to well represent local features of each image.

1.3.2 Point of focus

In this dissertation, the focus is on the phenomenon that atoms generated by learning dictionaries outperform DCT atoms in terms of feature expression diversity, and investigate whether this atom feature diversity can optimize existing compression methods in terms of image compression.

Recent research effort has been devoted to learning dictionaries that allow image coding to utilize adaptive transforms. As mentioned in subsection 1.2.2, one of the efficient methods to design such dictionaries is K-SVD. It is a technique that performs singular value decomposition on a particular set of matrices in order to design the most suitable atoms for the set. The design concept of K-SVD is to improve the sparsity of the transformation coefficients, or in other words, to minimize the number of non-zero coefficients under the condition that the data restoration accuracy is guaranteed. When applied to image compression, it is equivalent to reducing the number of coefficients that need to be coded while maintaining a certain image quality i.e. keeping the approximation error below a defined value. Given an image signal, K-SVD can derive a dictionary that well approximates each block with a sparse combination of atoms from the set of blocks composing the image.

It is known that dictionaries generated by K-SVD are largely dependent on the features of the training images. Therefore, the extension of K-SVD to support multiple dictionaries is a promising approach to more efficient representations of natural images with various

features. Here, let us call the extended K-SVD "multi-class K-SVD". Multi-class K-SVD adaptively selects the most suitable dictionary based on the local feature(s) of the image to be encoded. Therefore, the multi-class K-SVD dictionary will enable more efficient image representation than the traditional DCT dictionary.

In this thesis, I focus on three issues when applying a dictionary designed by multi-class K-SVD to image coding, that is, a multi-class dictionary design algorithm suitable for image coding, a sparse coefficient quantization technique that minimizes visual distortion, and an efficient entropy coding method for the quantized sparse coefficients.

1.3.3 Learning type base generation

In order to design multiple dictionaries, it is necessary to classify the learning data available in terms of characteristics. Various approaches such as edge directionality and pixel value variance have been studied as local features suitable for classification. As examples for image coding application, some classification methods based on intra prediction mode and intra/inter prediction residual power of the coding unit in H.264/AVC or H.265/HEVC have already studied.

It has been clarified that multi-class K-SVD gives better coding performance than single class K-SVD (i.e. with one dictionary). However, conventional studies on multi-class K-SVD use predetermined classification schemes for dictionary design, and do not consider the relationship between the classification stage and the dictionary training stage. Therefore, there still remains the potential for improvements in coding efficiency by combining dictionary training and classification optimization.

1.3.4 Quantization and code assignment

When applying the dictionaries that enable sparse representation to image compression, it is necessary to quantize the non-zero weighting coefficients, which are distributed in sparse, according to the target compression rate. Quantization is a lossy process, and the method of determining the quantization width for each sparse coefficient has a great effect on image quality. In the traditional international standards, the quantization width is determined by using the "quantization matrix" designed by the frequency characteristics of the atoms in DCT dictionary. However, the atoms included in the dictionaries designed by K-SVD are not composed of regular frequencies like FFT and DCT, and have complicated characteristics in which various frequency components are mixed. It has not yet been clarified what kind of quantization width should be used to quantize the weighting coefficients cor-

responding to such complicated atoms.

In addition, it is necessary to assign a code to each quantized sparse coefficient before transmission. Since DCT is used for image transform in the traditional international standard, non-zero coefficients have the characteristic of concentrating on the low frequency atoms with high probability. Utilizing this characteristic, various methods based on zigzag scan have been applied as the conventional code assignment method. However, since the atoms consisting of complex frequency components are randomly arranged in the dictionary designed by K-SVD, the conventional code assignment method cannot be applied as it is. In sparse representation coding, it is expected to develop a new code assignment method that utilizes the characteristic that the non-zero coefficients to be encoded are extremely distributed in sparse.

1.4 Structure of This Thesis

This thesis is structured as described below.

We start with an introduction to related work in Chapter 2. This includes an introduction to the overall framework of image coding technology and the key techniques including intra-frame/inter-frame prediction, DCT, KLT and Wavelet. Also, we introduce a review of the international standards for image coding such as JPEG and H.265/HEVC and their approach to image quality assessment.

Chapter 3 introduces the sparse representation of images. We will describe the dictionary design concept by K-SVD and its detailed algorithm, and explain the characteristics of the designed dictionaries by the simulation results of testing with some images. We also show that the image representation performance of the designed dictionaries is superior to that of the DCT dictionary, and mention the need for an approach with multi-class dictionary.

In Chapter 4, we propose a new method for designing a multi-class dictionary using K-SVD, and show its effectiveness by experiments. First, we introduce a conventional multiple dictionary design algorithm that uses classification based on the pre-determined fixed feature. Next, we propose a new multi-class K-SVD that obtains the optimum dictionary by alternately repeating the class update stage and the dictionary update stage. Then, we show a method for applying the designed multiple dictionaries to image compression. Finally, we will evaluate the performance of the proposed method by simulation experiments.

The dictionary generated by the method proposed in Chapter 4 enables sparse representation of images. That is, the transformation coefficient matrix is a sparse matrix with

very few nonzero values. For transmission, it is necessary to assign binary codes to the sparse coefficients for transmission. The main focus of Chapter 5 is to study how to assign the binary code. To clarify that, we analyze the statistical properties of the sparse coefficients in detail, and propose an entropy coding scheme that minimizes the amount of codes, and confirm the effectiveness by simulation experiments.

In addition to the encoding method, another topic worth discussing for sparse coefficients is how to quantize them. We will discuss the issue in Chapter 6. In this section, we will give a detailed description of human visual properties, and analyze the atoms in the designed dictionaries from a point of frequency characteristics. Based on the analysis, we show how to design a quantization matrix that adaptively quantizes the coefficients corresponding to each atom, and verify the effectiveness from the viewpoint of decoded image quality by experiments.

In Chapter 7, we will summarize the entire dissertation and contribution to the related research area, and present the direction of the future study on this topic.

Overview of Image Coding Technology 2

Contents

2.1 Basic Configuration of Video Coding

The core idea of video and image compression technology is to deal with a large number of "repetitive, correlated" elements (pixels or blocks) in the visual signal, which are considered to constitute information redundancy, and to minimize or eliminate them through various coding tools, that is, the working mechanism of compression algorithms.

The basic framework of the current video coding technology has been broadly formed in the early 1990s. It can be summarized as a combination of motion compensation, intra-frame prediction, inter-frame prediction, transformation, quantization, entropy coding, and image pre-processing such as RGB→YUV/YCbCr color conversion, noise reduction, image segmentation. Figure 2.1 shows the block diagram of the framework.

Furthermore, the design of the preprocessing part and the coding control functions which are not shown in the figure are outside the framework of standards, and they are left to the designer of the codec as an important part of the visual optimization.

The basic structure of almost video coding standards follows to this flow, but with the improvement of IT technology and hardware architecture, each coding tool can be more properly optimized in each application, which makes it possible to transmit video with higher resolution and compression rate.

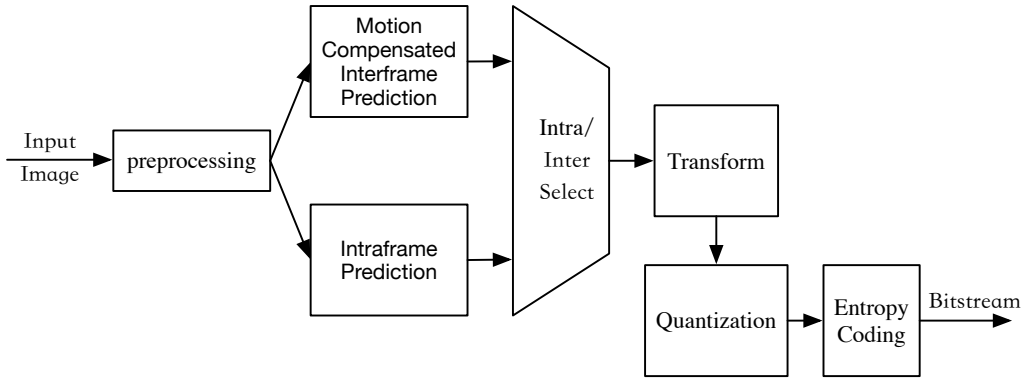


Figure 2.1: Block diagram of the basic framework of traditional video compression coding

2.2 Key Technologies

Generally, a digital image signal can be regarded as a two-dimensional (or in the case of video images, a three-dimensional) array consisting of pixel values of sampling points evenly spaced in the horizontal and vertical directions.

Almost of images we usually handle have a property that the pixel values at close distances are similar to each other, and the image arrays contain many redundant information.

Removing such redundancy from the original signal and transforming it into information that can be presented with a smaller number of bits is an essential operation for efficient image coding. Prediction is one of the key technologies to reduce such redundancy in image signals.

2.2.1 Prediction

The simplest prediction method for the next pixel value x is to use its previous pixel value x_{n-1} as the predicted value. Isn't it more efficient to encode the prediction error $e = x_n - x_{n-1}$, which is the difference between the predicted pixel value and the original pixel value, than to encode the original pixel value directly?

Although it is necessary to discuss the method of entropy coding as well, it is generally known that the smaller the variance of the target signal, the lower the amount of codes can be. In other words, if the neighboring pixels are similar to each other, the prediction error of the previous value will be close to zero, and the variance can be expected to be sufficiently small.

The coding method based on forward prediction was widely used in early video coding

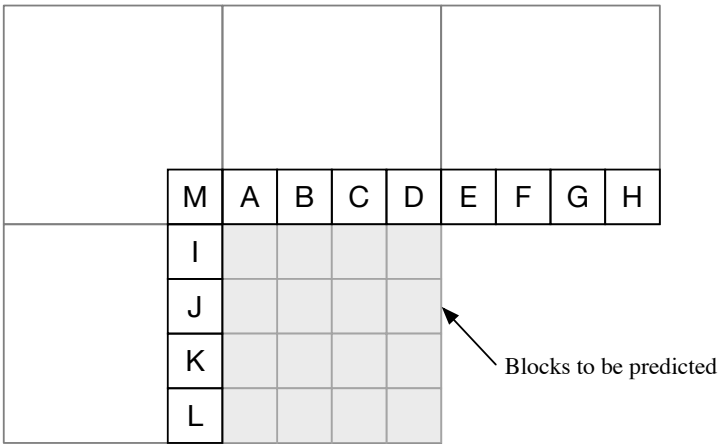


Figure 2.2: Placement of reference pixels used for 4×4 intra-prediction

schemes. Since image signals show a strong correlation not only in the horizontal direction but also in the vertical direction, two-dimensional prediction, which utilizes the decoded values in the upper scan line in addition to the previous value, is now common. Thus, a two-dimensional prediction method that encodes the difference between the predicted and predicted values in pixel units is called Differential Pulse Code Modulation (DPCM)[21, 22].

The orthogonal transformation coding, such as DCT, is widely used as a basic algorithm for non-reversible coding schemes because it is generally considered to have better performance than DPCM at low bit-rates[23]. However, since the orthogonal transformation is an independent block-by-block process, although it is effective in reducing the redundancy in the block, it has a disadvantage that the correlation between the blocks cannot be used simply in principle. For this reason, the combination of orthogonal transformation and intra-frame/inter-frame prediction of each block is also attracting attention.

2.2.1.1 Intra-frame prediction

Intra-frame prediction is the term for a prediction method used for in-frame coding without reference to other frames in the video image coding. It is also called "spatial prediction" because it uses the correlation of pixel values in the spatial direction.

The H.264/AVC[24], which has been designed to be a highly efficient video coding standard, has been refined to predict all pixels in the block before the orthogonal transformation, and multiple prediction modes are provided to support edges and textures in various directions.

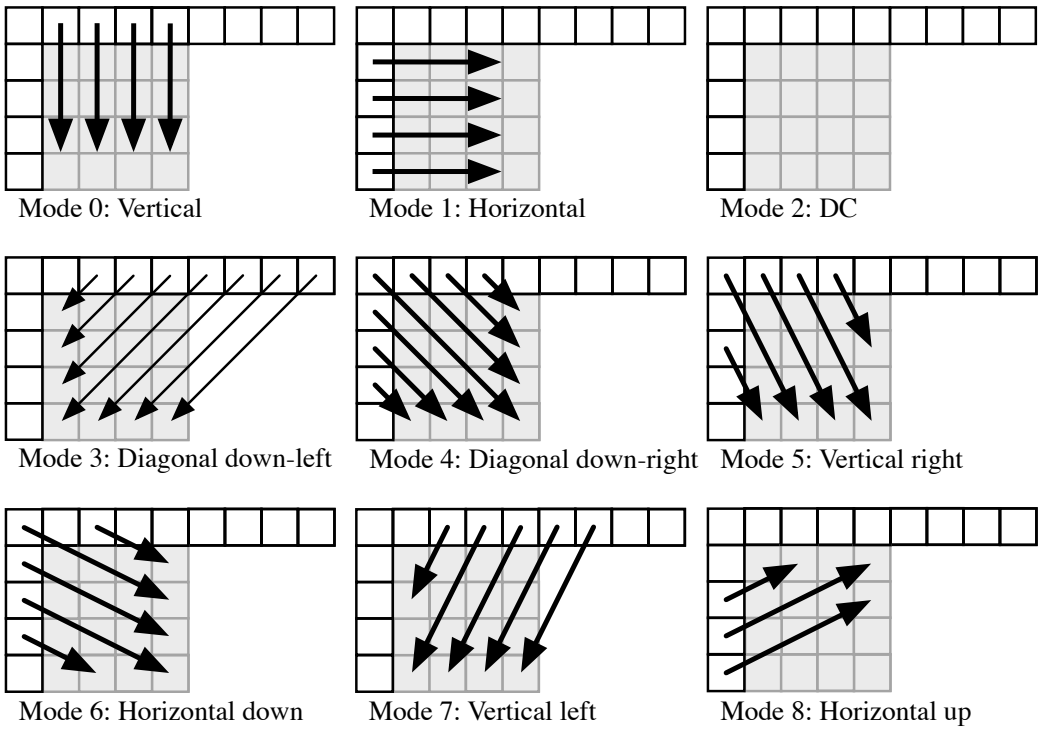


Figure 2.3: Prediction modes

For the Intra Prediction of 4×4 blocks, as shown in Figure 2.2, the prediction values are calculated by referring to a total of 13 pixels from A to M located at the boundaries of the surrounding encoded blocks. As shown in Figure 2.3, there are 9 prediction modes that define the method of calculating the prediction values (There are 9 prediction patterns for AVC, 35 prediction patterns for HEVC[25], and 89 prediction patterns for the upcoming VVC[26]), and the appropriate mode for each block can be selected by adaptation. For example, in the DC prediction of Mode 2 as shown in Figure 2.2, the average of the eight pixels from A to D and from I to L is obtained and the average is given as the predicted value for each of the 4×4 pixels. Since the predictions are uniform within a block, this prediction mode is suitable for flat areas of the image. Other prediction modes use the values of the reference pixels in the direction of the arrows in Figure 2.3 and are effective when there are textures with strong directional characteristics such as edges in each direction. As for which prediction mode is selected, it must be encoded as additional information for each block, but in the case of the same prediction mode as that of the adjacent block, only a small amount of code is required.

2.2.1.2 Inter-frame prediction

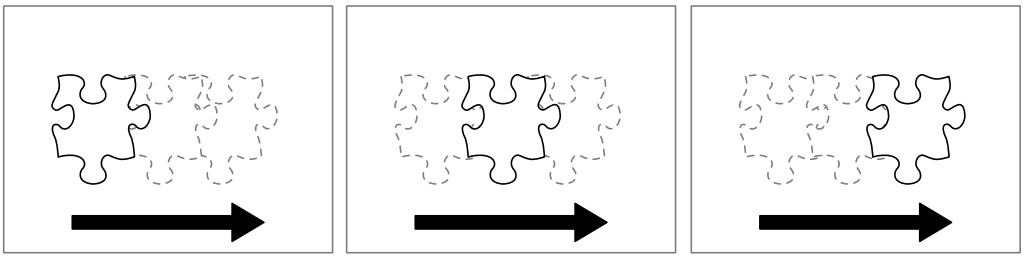
Video images used in television consist of about 30 frames per second, and there is a tendency that the correlation between consecutive frames is high. In video image compression, we consider the similarity between consecutive frames in the time dimension as such time redundancy. To simplify the understanding, we consider the case where the object is captured by a fixed camera (the background does not change).

The basic process is to compare the reference frame (previous frame) with the target frame (current frame). When there is no movement of the subject, the signal residual becomes zero when the difference is taken between the two frames. This process is called inter-frame prediction. On the other hand, if there is a movement of the subject, when the subtraction between frames is performed in the above manner, the difference data between the position the object was at in the previous frame and that of the current frame is generated and the rest of the frame becomes zero. If this situation is not solved, the difference data for two object shape regions are generated. To further reduce the amount of residual information, the following process is considered. First, the motion vector of the moving object is calculated between the current frame and the reference frame (previous frame). Next, the position of the object in the reference frame is shifted according to the motion vector to generate a motion compensated prediction image (predicted frame). Then, the difference between the motion compensated prediction image and the current frame image is calculated.

There are two types of inter-frame prediction methods: unidirectional prediction which uses predictions in one direction and bi-directional prediction. The pre-diction from the previous frame to the current frame is called forward prediction and the prediction from the future frame to the current frame is called backward prediction. When both of them are used, it is called bi-directional prediction, and the accuracy of the prediction is enhanced compared to unidirectional prediction. An example of inter-frame prediction is shown in Figure 2.4. For example, the prediction from (a) to (b) is forward prediction, and the prediction from (c) to (b) is backward prediction, and both of them are used in bi-directional prediction.

2.2.2 Transform

Compared with text data, image data is extremely large. Therefore, in image coding, it is important to develop a system with high coding efficiency, i.e., the data size for storage and transmission is small while maintaining image quality. Therefore, for image coding, linear



(a) previous frame (b) current frame (c) future frame

Figure 2.4: Prediction modes

transformations are used to reduce the amount of data for luminance, chrominance and their prediction error signals. Linear transformations for image coding include Discrete Cosine Transform (DCT)[27], which is the traditional standard, Discrete Wavelet Transform (DWT)[28], which does not cause block distortion in principle, and Karhunen-Loeve Transform (KLT)[29], which aims at least-square optimization, are well known as linear transformations for image coding.

2.2.2.1 KLT

KLT is a method of transform coding that transforms data into a more compressible form and removes redundancy caused by data correlation. KLT is also commonly referred to as Principal Component Analysis (PCA)[29] or Eigenvalue Decomposition (EVD). Their descriptions may vary somewhat in different disciplines, but the KLT in image compression that will be discussed here refers to methods that compute image covariance matrices, retaining larger eigenvalues and their corresponding eigenvectors.

The image compression procedure using KLT is as follows:

1. Data preparation

Firstly, the image to be processed will be split into N blocks of $\boldsymbol{M} \times \boldsymbol{M}$ size, and the data in the image blocks will be rearranged into a column vector in the order of left to right and top to bottom, The dimension of each column vector is $\boldsymbol{M} \times \boldsymbol{M}$. There are a total of N such column vectors, and the i-th column vector is $\boldsymbol{x}_i$.

Taking 2D vectors as an example, assuming that each vector can be represented as $[\boldsymbol{x}_1, \boldsymbol{x}_2]^T$, then these data points can be represented in the 2D plane, and suppose they are distributed in the ellipse shown in Figure 2.5.

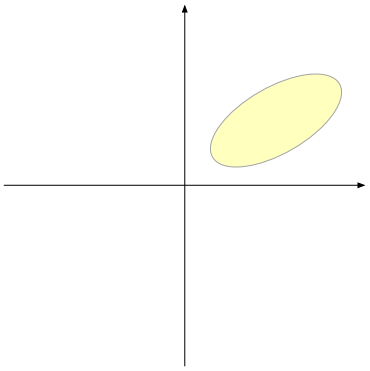


Figure 2.5: Example of a two-dimensional distribution of data

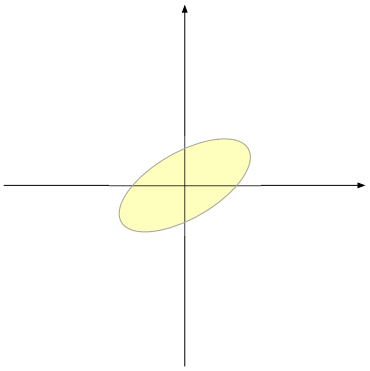


Figure 2.6: Data centering

2. Data centering

Let $\boldsymbol{x} = \boldsymbol{x}_i - \overline{\boldsymbol{x}}$, where $\overline{\boldsymbol{x}}$ is the mean vector of all column vectors, which can be seen as shifting the center of the dataset to the origin, shown in Figure 2.6.

3. Calculating the covariance matrix

We denote the covariance matrix by $\boldsymbol{C}$, then:

$$\begin{aligned} \boldsymbol{C} &= E[\boldsymbol{x}\boldsymbol{x}^T] \\ &= E[(\boldsymbol{x} - \overline{\boldsymbol{x}})(\boldsymbol{x} - \overline{\boldsymbol{x}})^T] \\ &= \frac{1}{N-1}\sum_{i=1}^{N}(\boldsymbol{x} - \overline{\boldsymbol{x}})(\boldsymbol{x} - \overline{\boldsymbol{x}})^T \end{aligned} \tag{2.1}$$

where $\overline{\boldsymbol{x}} = \frac{1}{N-1}\sum_{i=1}^{N}\boldsymbol{x}_i$.

4. Calculating the eigenvalues and eigenvectors of the covariance matrix

Denotes eigenvalues as $\lambda_1, \lambda_2, \ldots, \lambda_K$, and the corresponding eigenvectors are $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_K$ respectively, where $K = \boldsymbol{M} \times \boldsymbol{M}$. The covariance matrix $\boldsymbol{C}$ can then be expressed as:

$$\boldsymbol{C} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^T \tag{2.2}$$

in which, $\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_K]$ and $\boldsymbol{D} = diag(\lambda_1, \lambda_2, \ldots, \lambda_K)$

5. Decomposing and reconstructing vectors

After obtaining the covariance matrix and the eigenvector, it is possible to do a decomposition of the original vector, where the vector $\boldsymbol{x}$ can be expressed as a linear sum of the eigenvectors with the following mathematical expression:

$$\boldsymbol{x} = \sum_{i=1}^{K} c_i \boldsymbol{p}_i \tag{2.3}$$

where $c_i = \boldsymbol{p}_i^T \boldsymbol{x}, i = 1, 2, \ldots, K$.

Based on the above equation, by retaining the larger eigenvalues and their corresponding eigenvectors and discarding the smaller eigenvalues, the original vector can be approximately reconstructed, which result in dimension reduction or compression.

The approximation of $\boldsymbol{x}$ is denoted by $\hat{\boldsymbol{x}}$ and can be represented by the following equation.

$$\hat{\boldsymbol{x}} = \sum_{i=1}^{r} c_i \boldsymbol{p}_i, \quad r < K \tag{2.4}$$

As shown in Figure 2.7, retaining c_1 and discarding c_2 allows one parameter to express the information of the original two and achieve smaller errors (losing information in the $\boldsymbol{p}_2$ direction instead of the $\boldsymbol{x}_1$ or $\boldsymbol{x}_2$ direction).

2.2.2.2 Wavelet

The Wavelet Transform can be understood in conjunction with the Fourier Transform. The Fourier Transform uses a series of sine and cosine functions of different frequencies to decompose the original function, and the transformation yields the coefficients to form the original function using various frequencies of the sine and cosine. Similarly, the Wavelet Transform uses a series of wavelets of different scales to decompose the original function,

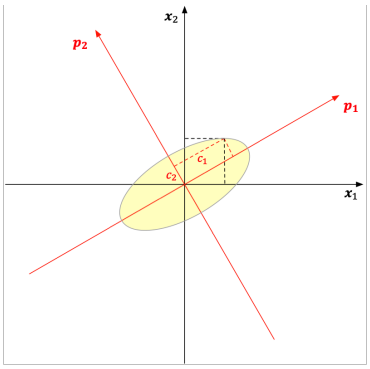


Figure 2.7: KLT of data

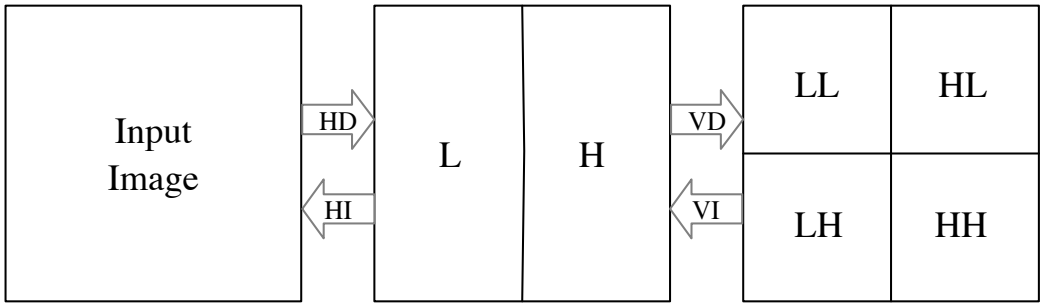
and the obtained coefficients form the original function using various scales of the wavelets. Different wavelets are decomposed by shifting and scaling, where shifting is used to get the time characteristics of the original function, and scaling is used to get the frequency characteristics of the original function.

The Fourier transform of an image is the decomposition of the image signal into sine-waves of various frequencies. Similarly, the Wavelet Transform is the de-composition of the image signal into a set of wavelets that are shifted and scaled by the original wavelet. In the image processing world, wavelets are called image microscopes because of their multi-resolution decomposition ability to decompose and peel away image information layer by layer. This is achieved by using low-pass and high-pass filters.

An example of the decomposition and reconstruction process of the two-dimensional discrete wavelet transform (DWT) for image coding is shown in Figure 2.8(a).

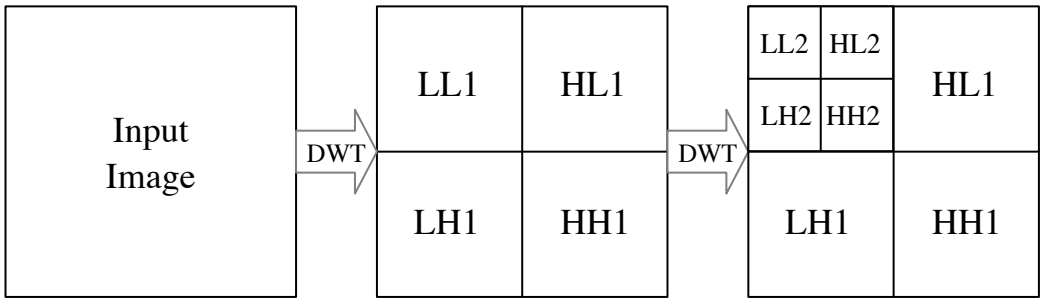
The decomposition process can be described as follows. Firstly, 1D-DWT is performed on each row of the image to obtain the low-frequency component L and the high-frequency component H of the original image in the horizontal direction. The low frequency component L is obtained by 2:1 subsampling in the horizontal direction after low-pass filtering, and the high frequency component H is obtained by 2:1 subsampling in the horizontal direction after high-pass filtering. Then, the vertical 1D-DWT is performed on the transformed data to obtain LL with the low-frequency component in the horizontal/vertical directions, LH with the horizontal low-frequency and the vertical high-frequency, HL with the horizontal high-frequency, and the vertical low-frequency, and HH with the high-frequency component in both horizontal/vertical directions.

The reconstruction process can be described as follows. Firstly, a one-dimensional inverse discrete wavelet transform is performed on each column of the transformation result,



HD: **h**orizontal **d**ecomposition VD: **v**ertical **d**ecomposition
HI: **h**orizontal **i**nverse VI: **v**ertical **i**nverse

(a) Decomposition and reconstruction process of 2D-DWT



(b) 2 level decomposition of 2D-DWT

Figure 2.8: Block diagram of 2D-DWT for image coding

and then a one-dimensional inverse discrete wavelet transform is performed on each row of the data obtained from the previous step. As the result, we can obtain a reconstructed image. The complete reconstruction is possible by using Haar filter, SSKF, Daubechies filter, etc. as low-pass filter and high-pass filter.

From the above processes, the wavelet decomposition of the image is a process of separating the original signal to a low frequency component and multiple directional high frequency components. In the process of decomposition, the LL component can be further wavelet decomposed as needed until the requirements are met, as shown in Figure 2.8(b).

2.2.2.3 DCT

Discrete Cosine Transform (DCT) is similar to the Discrete Fourier Transform (DFT)[30], but uses only real numbers. Among the general orthogonal transformations of speech and image signal transformations, the DCT is considered as a quasi-optimal transform. In a series of international standard recommendations for video compression coding issued in

recent years, DCT has been used as one of the basic processing modules.

DCT is often used for lossy data compression of voice, audio, image and video signals. This is due to the strong "energy concentration" of the discrete cosine transform: most of the energy of natural signals (such as sound and images) is concentrated in the low-frequency part of the discrete cosine transform, and the de-correlation performance of the DCT is close to that of the KLT when the signal has statistical properties approximating those of a Markov process.

The principles of DCT are described as follows:

1. 1D-DCT

There are a total of 8 forms of 1D DCT, of which the following is the most common one:

$$F(u) = c(u) \sum_{i=0}^{N-1} f(i) \cos\left(\frac{(2i+1)\pi u}{2N}\right) \tag{2.5}$$

in which,

$$c(u) = \begin{cases} \sqrt{\frac{1}{N}} & u = 0 \\ \sqrt{\frac{2}{N}} & u \neq 0 \end{cases} \tag{2.6}$$

where N is the total number of elements of the input one-dimensional data, and the coefficient $c(u)$ makes DCT matrix an orthogonal matrix.

2. 2D-DCT

In the above equations, transformation of one-dimensional signals has been discussed. However, the image is a two-dimensional signal whose values are arranged vertically and horizontally. In other words, the results of 2D transformations are obtained by applying 1D transformations to all the rows, and then performing 1D transformations on all the resulting columns. This process is represented by a single expression:

$$F(u,v) = c(u)c(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) \cos\left[\frac{(2i+1)\pi u}{2N}\right] \cos\left[\frac{(2j+1)\pi v}{2N}\right] \tag{2.7}$$

where

$$c(u) = \begin{cases} \sqrt{\frac{1}{N}} & u = 0 \\ \sqrt{\frac{2}{N}} & u \neq 0 \end{cases}, \quad c(v) = \begin{cases} \sqrt{\frac{1}{N}} & v = 0 \\ \sqrt{\frac{2}{N}} & v \neq 0 \end{cases} \tag{2.8}$$

In practice, it is common to discuss the situation where two N are equal, i.e. the data is in the form of a square matrix, which can be written in matrix form as follows:

$$\begin{aligned} F =& AfA^T \\ A_{u,v}(i,j) =& \cos\left[\frac{(2i+1)\pi u}{2N}\right]\cos\left[\frac{(2j+1)\pi v}{2N}\right] \end{aligned} \tag{2.9}$$

where A is denoted as the base of the 2D-DCT. The complexity of 2D-DCT reaches $O\left(n^4\right)$, so the matrix for DCT transformation should not be too large. In the practical process of image manipulation, the matrix needs to be divided into blocks, usually into 8×8 or 16×16 size, so that the DCT transformation does not take too much time.

The formula for the 2D inverse DCT (2D-IDCT) is as follows:

$$f(i,j) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} c(u)c(v)F(u,v)\cos\left[\frac{(2i+1)\pi u}{2N}\right]\cos\left[\frac{(2j+1)\pi v}{2N}\right] \tag{2.10}$$

Where $c(u)$ and $c(v)$ are same as equation (2.8). And the matrix form transformation formula for 2D-IDCT is as follows:

$$f = A^{-1}F(A^T)^{-1} = A^TFA \tag{2.11}$$

As shown in Figure 2.9, the original 8×8 pixels are transformed into another 8×8 array of coefficients by DCT. And, the original 8×8 pixels is presented by weighted sum of the DCT base functions, in which weighted value is a coefficient corresponding to each base. In the coefficient matrix, the low frequency energy of the original image corresponds to the upper left corner of the matrix and the high frequency energy corresponds to the lower right corner of the matrix. Therefore, considering the statistical property of the image that the correlation between neighboring pixels is high, the coefficients with large absolute values are concentrated in the left corner. When u, v is 0, $F(0,0)$ in the upper left corner is a mean value of all pixels, called the DC component, or DC coefficient, and as u, v increases, the rest of the matrix is the AC component, or AC coefficient. It also shows that the original image is represented by the linear sum of the basis, i.e. $f\left(i,j\right) = \sum_u\sum_v F(u,v)A_{uv}(i,j)$.

2.3 International Standards for Image Coding

The basic structure of the international standard was already settled in the early 1990's. It consists of Y/Cb/Cr color representation, entropy coding, frequency transformation and

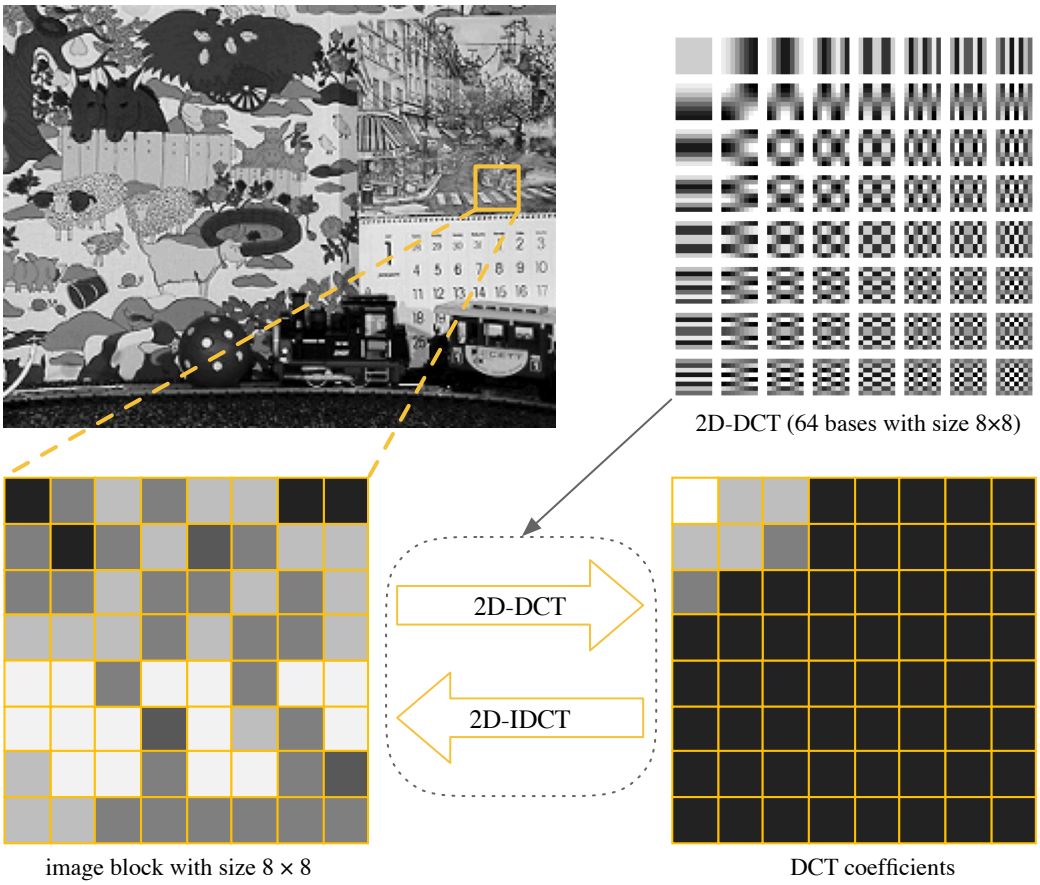


Figure 2.9: 2D-DCT & IDCT in practical procedures

quantization for both still and moving images, and coding tools such as intra-prediction, inter-prediction and intra-loop filtering for moving images.

In this section, one representative international standard will be presented for still images and video compression, respectively.

2.3.1 JPEG

JPEG (Joint Photographic Experts Group)[31, 32] is an image compression format used in almost all digital cameras. It is still used in a wide range of applications due to its good balance of performance and processing load, even though its successors, JPEG 2000[31] and JPEG XR[31], have been developed. JPEG is lossy compression that takes advantage of the properties of the human visual system, using a combination of quantization and lossless compression coding to remove redundant information from the perceptual and statistical

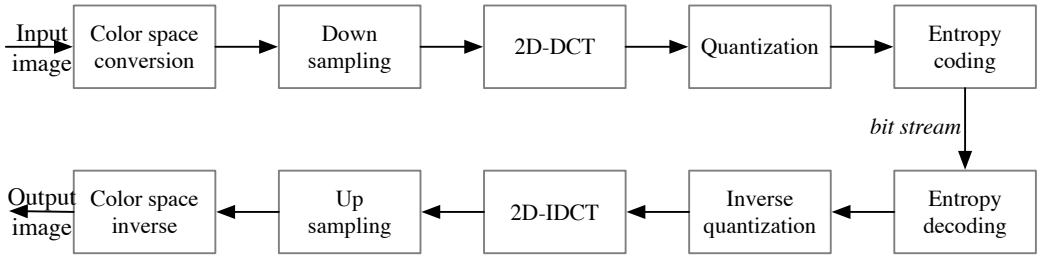


Figure 2.10: Block diagram for JPEG encoding and decoding

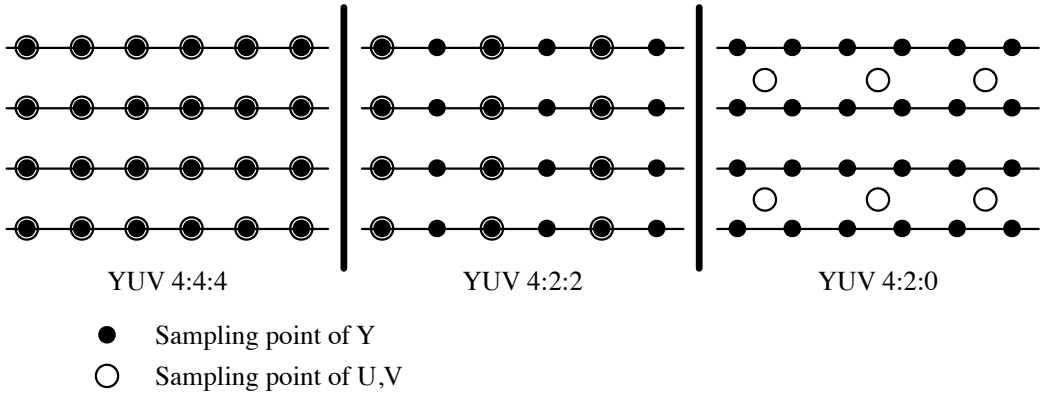


Figure 2.11: Three sampling methods used in JEPG

viewpoints. Its encoding and decoding process is shown in Figure 2.10.

1. Color space conversion

First, the video is converted from RGB (red, green, and blue) to a different color space called YUV (the Y component represents the brightness of a pixel, and the U and V components together represent hue and saturation). This coding system is useful because the human eye is more sensitive to luminance differences than to color variations. Based on this knowledge, encoders can be designed to compress images more efficiently. Since U and V have narrower bandwidths, reducing the image size to half Y does not significantly affect human perception.

2. Down sampling

The conversion made above makes the next step possible, which is the reduction of the U and V components (also called "chroma subsampling"). The ratio of this chroma subsampling on JPEG can be 4:4:4 (no chroma subsampling), 4:2:2 (a multiple of 2 in the horizontal direction), and the most common 4:2:0 (a multiple of 2 in the horizontal and vertical

directions), as shown in Figure 2.11. For the remainder of the compression process, Y, U, and V are all treated individually in a very similar manner.

3. Discrete cosine transform

Next, each component of the video (Y, U, V) is generated into three regions, each of which is subdivided into tile-like blocks with 8×8 pixels, which are trans-formed into frequency space using a two-dimensional discrete cosine transform. This is done by subtracting 128 from each value in the block to make its range $[-128, 127]$, and then using the discrete cosine transform and rounding to obtain the result. The significant value in the upper left corner of the resulting 8×8 coefficients is called the DC coefficient; the other 63 values are called AC coefficients. The DC coefficients in all 8×8 blocks are then processed using DPCM[22], and the AC coefficients in each block are processed using Run Length Encoding (RLE)[33].

4. Quantization

The human eye readily recognizes subtle differences in brightness over a relatively large area, but has difficulty discerning the exact intensity of high-frequency brightness variations. This gives us an excellent reduction in the amount of information at the higher frequency components. The main lossy operation in the process is to simply divide each component in the frequency domain by a constant corresponding to the component, and then round to the nearest integer. With this result, many of the higher frequency components are often rounded to near zero, while many of the remaining components become small positive or negative numbers. A general quantization matrix is:

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Each component in the matrix of DCT coefficients is divided by the corresponding element in the quantization matrix.

5. Entropy coding

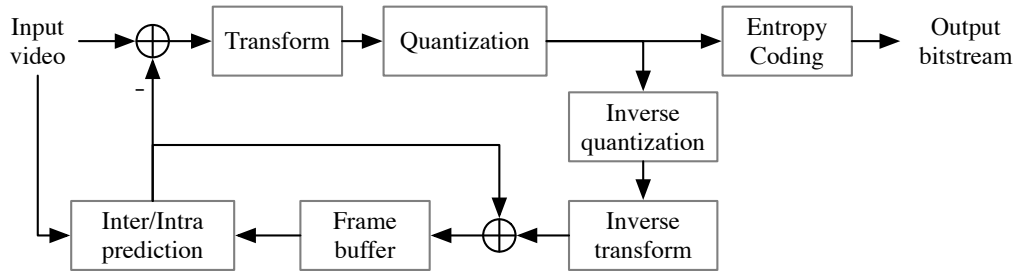


Figure 2.12: Typical hybrid video coding architecture

The first step of entropy coding is to scan the elements of the quantized DCT coefficient matrix from the low frequency coefficient in the upper left to the high frequency coefficient in the lower right in a zigzag manner to make them one-dimensional. Then, for the one dimensional data, the Huffman code[34] is assigned to (R, L), which is a pair of zero-run length R and subsequent non-zero coefficient level L. At the beginning part of the scan, R tends to be small and L tends to be large, while at the end part of the scan, R tends to be large and L tends to be small. Taking advantage of this probability characteristics, an efficient Huffman code table has been designed. In addition, a special symbol called EOB (End of Block) is provided for long zero runs that continue to the end of the scan. The JPEG standard also allows the use of arithmetic encoding[35] that is mathematically superior to the Huffman encoding. The use of arithmetic encoding generally makes the file about 5% smaller.

At this point the JPEG compression encoding is complete, and to use the compressed image it needs to be decoded, i.e. all the above operations in reverse.

2.3.2 H.265/HEVC

High Efficiency Video Coding (HEVC), also known as H.265 and MPEG-H Part 2, is a video compression standard that is considered a successor to the ITU-T H.264/MPEG-4 AVC standard. Its development began in 2004 by the ISO/IEC Moving Picture Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG). The first version of the HEVC/H.265 video compression standard was accepted as an official standard by the International Telecommunication Union (ITU-T) on April 13, 2013, and it is considered to achieve twice the compression efficiency of H.264/MPEG-4 AVC, that is, which equates to a 50% reduction in bit rate for the same picture quality. It can support 4K definition and even up to Ultra High Definition TV (UHDTV) with a maximum definition of 8192×4320 (8K definition)[25, 36].

Like H.264/AVC and many other video compression codecs, HEVC is based on the hybrid video coding architecture (see Figure 2.12), but with some new techniques added to each part or to improve the efficiency of the original coding tool. The introduction of the various coding features used in HEVC hybrid video coding is given below.

1. Coding Tree Unit and Coding Tree Block structures

In previous standards, such as H.264/AVC, the unit of the encoding layer is the macro-block, which in the conventional 4:2:0 color sampling format contains one 16×16 luminance block and two corresponding 8×8 chroma blocks, while a similar core structure in HEVC is called the Coding Tree Unit (CTU), whose dimensions is determined by the encoder and can be larger than traditional macro-blocks. A CTU contains one luminance Coding Tree Block (CTB) and the corresponding multiple chroma CTBs, as well as other syntactic elements. The size of the luminance CTBs can be expressed as L $\times$ L, where L = 16, 32 or 64, the larger the L the more efficient the compression. HEVC also supports splitting the CTBs into smaller blocks using a tree structure and quad-tree-like signaling.

2. Code Unit (CU) and Code Block (CB)

The syntax of the CTU quadratic tree illustrates the size and location of each luminance and chrominance CB. The root node of the quadratic tree represents the CTU. therefore, the size of the luminance CTB is also the maximum size of the luminance CB. The way to split the CTUs into luminance CBs and chrominance CBs is expressed together. A luminance CB usually corresponds to 2 chrominance CBs, and the data from all three CBs, along with the associated syntactic rules, form a coding unit. A CTB can contain only one CU, or it can be divided into several CUs, each of which has a corresponding division method to indicate how to divide it into prediction unit (PU), and each CU also has a transformation tree to indicate how to divide it into transformation unit (TU).

3. Prediction unit (PU) and prediction block (PB)

The decision of whether the prediction takes an inter-frame or intra-frame mode is made at the CU level. The root of the PU segmentation structure is located on the CU level. According to the basic prediction mode decision, each luminance and chrominance CB of the CU is further segmented into various sizes and predicted using the corresponding luminance and chrominance PBs. HEVC supports various PB sizes from 64×64 to 4×4.

4. Transformation Unit (TU) and Transformation Block (TB)

The predicted residuals are encoded using a block transformation, and the root of the TU tree structure is located on the CU level. Both the luminance CB and the chrominance CB can be transformed as one TB overall or further split into smaller TBs. 4 types of TB squares are supported by HEVC: 4×4, 8×8, 16×16 and 32×32, for each of which an integer base function similar to the discrete cosine transform is defined. For the residuals of the luminance component after intra-frame prediction, there is an alternative integer transformation scheme evolved from the discrete sine transform (DST) if the 4×4 transformation is performed.

5. Motion compensation

The highest precision of the motion vector (MV) is 1/4 pixel, and samples at non-integer pixel locations are interpolated using 7 or 8 taps filters. Same as H.264/MPEG-4 AVC, HEVC also uses multi-reference frame mechanism. Corresponding to unidirectional and bidirectional prediction, each PB can have 1 or 2 motion vectors.

Video compression coding is a very active area of research, although the latest coding standard HEVC compared to the previous generation of coding standards has shown great performance improvements, Joint Video Experts Team (JVET)'s early exploratory work to prove that it is possible to study compression coding methods that outperform HEVC. A series of Core Experiments are currently underway to explore coding performance, coding complexity, etc., and emerging technologies such as deep learning are also being applied to video coding. The industry is currently devising more efficient compression schemes for the current standard HEVC, which will provide immersive media formats such as VR and AR, where 8K sources are considered essential. The key to achieving this is the JVET and MPEG's proposed Versatile Video Codec (VVC), which, aims to achieve up to 50% compression efficiency compared to HEVC while maintaining video quality. The official version of the standard will be released in October 2020.

2.3.3 Coding efficiency evaluation

2.3.3.1 Image quality evaluation

Images are important information sources for human perception and machine pattern recognition, and their quality plays a decisive role in the adequacy and accuracy of the acquired information. However, images will inevitably be distorted in the process of compression and transmission. It is necessary to establish an effective image quality evaluation system to measure the image quality. Currently, image quality evaluation can be categorized into a

subjective evaluation method and an objective evaluation method, the former relying on the subjective perception of experimenters to evaluate the quality of the object; the latter based on the quantitative indicators given by the model, simulating the perception mechanism of the human visual system to measure image quality.

Subjective evaluation, which is easy to understand, is the evaluation of image quality according to the viewer's subjective perception of the image. A common procedure is to give the original image (reference image) and the distorted image (image to be evaluated), and ask the viewers to rate the distorted image, and then sum up all the subjective scores to get the Mean Opinion Score (MOS). In recent years subjective evaluation methods have been more widely used in the fields of neural network image recognition and style migration.

Objective evaluation of image quality is based on a mathematical model of the subjective visual system of the human eye, and the quality of the image is calculated through a specific formula. Compared to the subjective evaluation, the objective evaluation is characterized by batch processability and reproducibility of the results, so there will be no deviation due to human factors.

Objective evaluation algorithms can be divided into three categories according to their dependence on reference images.

1. full-reference: it requires a one-to-one comparison with a reference image;
2. semi-reference: it only requires a partial comparison with a reference image;
3. non-reference: it does not require a specific reference image.

The full-reference algorithm is the longest studied and most mature, and is the main way to evaluate the merits of compression algorithms.

The quality of the image signal to be evaluated can be analyzed by the quality of the error signal obtained after comparison with the original image signal. The degradation of the image quality is related to the power of the error signal. Based on this, the simplest quality evaluation algorithm is Mean Squared Error (MSE) and Peak Signal-Noise Ratio (PSNR). The expression are as follows.

$$\begin{aligned} \text{MSE} =& \frac{\sum_{m=1}^{M} \sum_{n=1}^{N} [R(m,n) - I(m,n)]^2}{M \times N} \\ \text{PSNR} =& 10 \log_{10} \frac{L_{\max}^2}{\text{MSE}} [\text{dB}] \end{aligned} \tag{2.12}$$

where $R(m,n)$ is the grayscale value of the reference image at spatial position (m,n), $I(m,n)$ is the grayscale value of the distorted image at spatial position (m,n), and $L_{\max}$ the peak signal, $L_{\max} = 255$ for an 8-bit grayscale image.

PSNR is the most widely used objective measurement method for evaluating image quality, but the actual test results show that the evaluation results of PSNR do not always reflect the evaluation results of the human eye. Images with low PSNR often feel visually worse than images with high PSNR. This is because the human eye's visual sensitivity to error is not absolute, and its perception of the results varies depending on many factors.

Natural images are highly structured, as evidenced by strong correlations between the pixels of an image, especially when they are spatially similar. These correlations carry important information about the structure of an object in a visual scene. Wang et al. proposed a structural distortion-based image quality evaluation method, called Structure-Similarity Index (SSIM)[37], which argues that illumination is independent of the structure of an object, and illumination changes mainly from luminance and illuminance changes. The luminance information of an object's surface is related to illuminance and reflection coefficient, and the structure of an object in a scene is independent of illuminance and reflection coefficient is related to the object. We can explore the structural information in an image by separating the effect of illuminance on the object. Here, the luminance and contrast, which are related to the structure of the object, are taken as the definition of the structural information in an image. Since the luminance and contrast in a scene are always changing, we can obtain more precise results by processing the localities separately.

The SSIM index models the distortion as a combination of three different factors: luminance, contrast, and structure. Using the average as an estimate of luminance, the standard deviation as an estimate of contrast, and the covariance as a measure of structural similarity. The mathematical derivation of SSIM is as follows.

$$\mu_X = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X(i,j) \tag{2.13}$$

$$\sigma_X = \left(\frac{1}{H \times W - 1} \sum_{i=1}^{H} \sum_{j=1}^{W} \left(X(i,j) - \mu_X \right)^2 \right)^{\frac{1}{2}} \tag{2.14}$$

$$\sigma_{XY} = \frac{1}{H \times W - 1} \sum_{i=1}^{H} \sum_{j=1}^{W} \left(X(i,j) - \mu_X \right) \left(Y(i,j) - \mu_Y \right) \tag{2.15}$$

$$l(X,Y) = \frac{2\mu_X\mu_Y + C_1}{\mu_X^2 + \mu_Y^2 + C_1} \tag{2.16}$$

$$c(X,Y) = \frac{2\sigma_X\sigma_Y + C_2}{\sigma_X^2 + \sigma_Y^2 + C_2} \tag{2.17}$$

$$s(X,Y) = \frac{2\sigma_{XY} + C_3}{\sigma_X^2 + \sigma_Y^2 + C_3} \tag{2.18}$$

$$\text{SSIM}(X,Y) = l(X,Y) * c(X,Y) * s(X,Y) \tag{2.19}$$

When $C3 = C2/2$, SSIM can be simplified to:

$$\text{SSIM}(X,Y) = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)} \tag{2.20}$$

2.3.3.2 Bjøntegaard metric

The Video Coding Experts Group (VCEG) recommended the use of the Bjøntegaard model to calculate the gain effect between two different coding methods[38, 39]. Because of the advantages of PSNR, such as ease of calculation, PSNR was also chosen to evaluate coding distortion in the very first Bjøntegaard delta (BD) metric model. Therefore, the BD metric model consists of two metrics: BD-rate and BD-PSNR. The former indicates the average percentage of bit rate that the comparison coding method can save over the reference coding method with the same objective quality. The latter shows the average difference in PSNR between the comparison coding method and the reference coding method, under the same bit rate.

Consider that on the rate-distortion (RD) curve, the range of high bit rate regions is larger than low bit rate regions for the same percentage. For example, the same 33% bit rate saving is four times larger in the range of 1500-2000 kbps than in the range of 375-500 kbps[40]. Therefore, the BD metric takes the logarithm on the bit rate axis of the RD curve plot.

In order to implement the BD measurement model, as shown in Figure 2.13, it is first necessary to evaluate the images processed by two different coding methods using the same evaluation method, and plot the evaluation results as RD curves. Then, the integration interval is confirmed, the interval in the direction of the bit rate axis is $[a, b]$, and the interval in the direction of the distortion rate axis is $[c, d]$, where a, b denote the lowest and highest values of the integration bound for the bit rate R, respectively. Similarly, c and d denote the

range of values of the distortion rate D, as

$$\begin{aligned} a &= \max\left(\min\left(\mathrm{R}[\mathrm{S}_1]\right), \min\left(\mathrm{R}[\mathrm{S}_2]\right)\right) \\ b &= \min\left(\max\left(\mathrm{R}[\mathrm{S}_1]\right), \max\left(\mathrm{R}[\mathrm{S}_2]\right)\right) \\ c &= \max\left(\min\left(\mathrm{D}[\mathrm{S}_1]\right), \min\left(\mathrm{D}[\mathrm{S}_2]\right)\right) \\ d &= \min(\max\left(\mathrm{D}[\mathrm{S}_1]\right), \max\left(\mathrm{D}[\mathrm{S}_2]\right)) \end{aligned} \tag{2.21}$$

It is noted that the BD measurement model does not use the measured data directly, but generates their polynomial fit functions $\mathrm{S}_1^{'}$,$\mathrm{S}_2^{'}$ from the RD curves in advance. Utilizing the fitted polynomial and calculating on the integral interval, the BD measurements can be obtained from the calculations according to the following equation.

$$\begin{aligned} \text{BD-PSNR} &= \frac{\int_a^b \mathrm{R}[\mathrm{S}_1^{'}] - \mathrm{R}[\mathrm{S}_2^{'}] d\mathrm{R}}{b-a} \\ \text{BD-rate} &= \exp\left\{\frac{\int_c^d \mathrm{D}[\mathrm{S}_1^{'}] - \mathrm{D}[\mathrm{S}_2^{'}] d\mathrm{D}}{d-c}\right\} - 1 \end{aligned} \tag{2.22}$$

BD-rate is usually expressed as a percentage relative to the reference curve (i.e. S_2 in Figure 2.13), so that a negative number represents a compression gain, while a positive number represents a compression loss.

2.4 Possibility of Base Expression Beyond DCT

From the point of view of signal representation by the bases, the original signal $\boldsymbol{y}$ can be represented in the form $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x}$, where $\boldsymbol{D}$ is the dictionary formed with the transformation bases and $\boldsymbol{x}$ is the corresponding coefficient vector. Here, it is desirable to be able to approximate the original signal as much as possible using as few coefficients as possible.

Thus, from the viewpoint of sparse representation, the compression problem can be transformed into an optimization problem for the sparse approximation model, the mathematical expression of which is as follows.

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0, \quad s.t. \quad \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|_2 \leq \varepsilon \tag{2.23}$$

where ε denotes the maximum allowable error, and $\|\boldsymbol{l}\|_p$ denotes the norm operation, which is equivalent to counting the number of non-zero values of vector $\boldsymbol{l}$ when p is equal to 0, and to calculating the Euclidean distance when it is equal to 2.

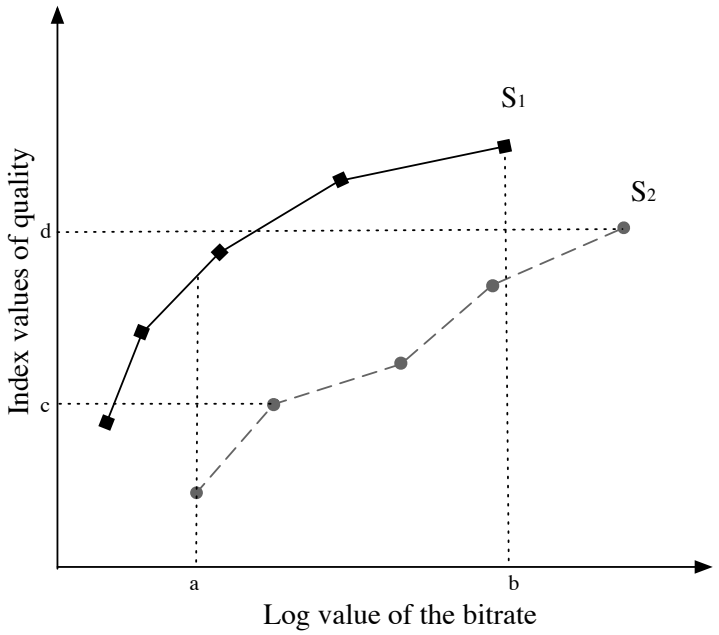


Figure 2.13: RD curve plot for two coding methods under the same quality evaluation model

The core issue of the optimization problem is the choice of dictionary. Most solution methods can be grouped into two categories, one of which can be categorized as methods based on decomposition ideas, such as the Fourier transform, discrete cosine transform, wavelet transform, and other methods chosen by traditional compression methods. The dictionaries constructed by this type of methods are usually highly structured and easy to design fast algorithms, however, because this type of approach requires a pre-defined mathematical model of the data, it is difficult to approximate the optimal solution for compressed objects that deviate from the pre-defined model.

In contrast to this, there is an approach based on machine learning to construct dictionaries. This type of approach does not pre-model the data structure, but instead uses a large number of training examples to generate dictionaries. Such dictionaries are usually expressed in the form of a matrix. Common dictionary learning methods include: maximum likelihood (ML)[41], MOD[42], PCA[29, 43], and K-SVD[18] , which will be explained in more detail later on. The disadvantage, however, is that the dictionaries themselves lack structure, so that they are more complex to apply. In addition, the representation performance for data with features that deviate significantly from the characteristics of the training data by the training-based dictionary is not as good as that by the dictionary based on the decomposition idea.

In summary, dictionaries based on the decomposition idea can show better expression performance for all data structures on average, but cannot achieve an optimal solution, while dictionary construction ideas based on machine learning may be able to achieve optimal expression for corresponding data, at the cost of reduced expressiveness for non-corresponding data. So, is it possible to make the two categories complement each other's strengths? If the image can be represented with a small number of non-zero coefficients, it may lead to information compression beyond DCT as a result.

2.5 Summary

In this chapter, we reviewed the basic architecture of currently widely used image coding technologies and introduced some key technologies included in encoder and decoder. The overall flow of mainstream video coding was first introduced in Section 2.1, and the key techniques in the flow: prediction and transformation were described in detail later in Section 2.2. The subsection on prediction introduced intra and inter-frame prediction, while the subsection on transformation introduced three classical transformation algorithms: KLT, DWT and DCT.

In section 2.3, the mainstream international standards, JPEG for still image compression and H.265/HEVC for video compression, were introduced, and the methods for evaluating image quality were explained, followed by a brief introduction of the MOC method for subjective evaluation , and the widely used objective evaluation methods, PSNR and SSIM, were also explained.

In Section 2.4, we illustrated the idea of understanding base transform processing in terms of sparse representation, and suggested the possibility of exceeding the compression efficiency of traditional DCT base transforms by improving the way dictionaries (set of transform bases) are generated.

Sparse Representation for Images 3

Contents

Figure 3.1 shows a concept of sparse representation. From the point of view of matrix factorization, as mentioned in Section 2.4, the dictionary learning process is equivalent to decompose a given sample dataset $\boldsymbol{Y}$ (each column of $\boldsymbol{Y}$ represents sample $\boldsymbol{y}^i$) into dictionary matrix $\boldsymbol{D}$ and coefficient matrix $\boldsymbol{X}$. That is,

$$\boldsymbol{Y} \approx \boldsymbol{D} * \boldsymbol{X} \tag{3.1}$$

where, satisfying the constraint that $\boldsymbol{X}$ is as sparse as possible. $\boldsymbol{D}$ is called a "dictionary", and each column of $\boldsymbol{D}$ is called a base or atom; $\boldsymbol{X}$ is called a coefficient matrix. In practical image processing, $\boldsymbol{Y}$ is commonly composed of blocks extracted from images with the same size. When the number of pixels in a block is n, the pixels in each block is reshaped into column vector of size n, as shown in Figure 3.2. If m training blocks are extracted from an image, the training data matrix $\boldsymbol{Y}$ with m columns $\times n$ rows is composed.

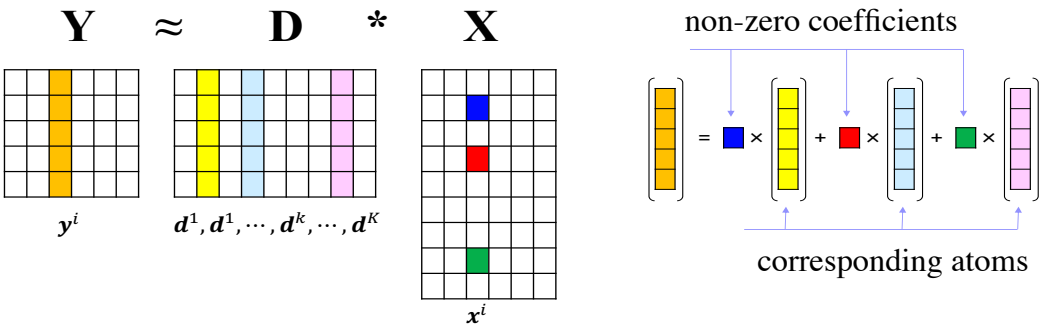


Figure 3.1: Illustration of sparse representation

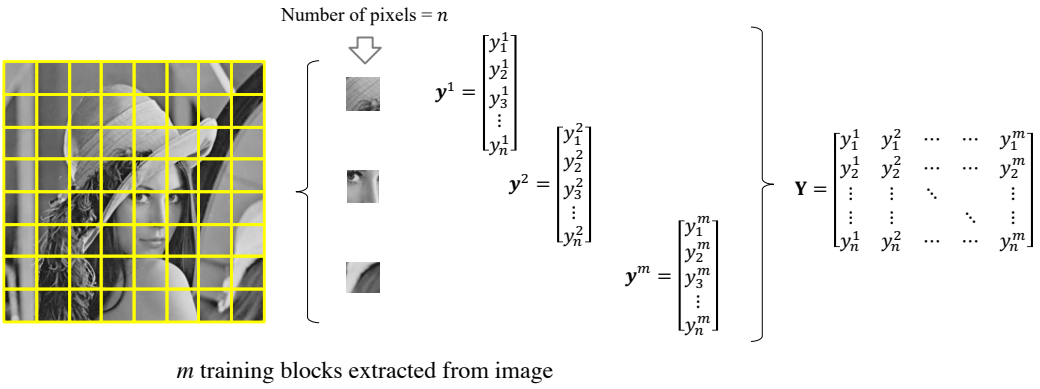


Figure 3.2: Illustration of training data $\boldsymbol{Y}$ obtained from images

Dictionary learning can have the following three objective function forms and they are equivalent to each other.

$$\boldsymbol{D}, \boldsymbol{X} = \arg\min \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|^2 + \lambda\|\boldsymbol{X}\|_0 \tag{3.2}$$

$$\boldsymbol{D}, \boldsymbol{X} = \arg\min_{\boldsymbol{D},\boldsymbol{X}} \|\boldsymbol{X}\|_0, \quad s.t. \quad \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|^2 \leq \varepsilon \tag{3.3}$$

$$\boldsymbol{D}, \boldsymbol{X} = \arg\min_{\boldsymbol{D},\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|^2, \quad s.t. \quad \|\boldsymbol{X}\|_0 \leq T_0 \tag{3.4}$$

where $\|\boldsymbol{X}\|_0$ is the number of non zero coefficients in the matrix $\boldsymbol{X}$. The first form is often approximated using the L_1 norm term instead of L_0 norm because it is difficult to solve. In the third form, T_0 is a constant value called the"sparsity constraint parameter".

It is clear that the problem faced by dictionary learning is difficulty to solve because the only known variable is $\boldsymbol{Y}$, and $\boldsymbol{D}$ and $\boldsymbol{X}$ to be solved are unknown.

K-SVD is a classical algorithm for dictionary learning to solve the above problem, and it alternately repeats updating the coefficient $\boldsymbol{X}$ under fixing the dictionary $\boldsymbol{D}$ and updating the dictionary $\boldsymbol{D}$ under fixing the coefficient $\boldsymbol{X}$.

In addition, if the number K of columns of $\boldsymbol{D}$, is less than the number n of rows of $\boldsymbol{Y}$, $\boldsymbol{D}$ becomes an under-complete dictionary, similar to dimension reduction by PCA; if K is exactly equal to n, then $\boldsymbol{D}$ is called a complete dictionary, for example DCT in JPEG; if K is greater than n, $\boldsymbol{D}$ is called an over-complete dictionary.

To illustrate the problems faced in learning over-complete dictionaries, suppose we have an $H * W$ over-complete dictionary $\boldsymbol{D}$ as the initial dictionary, a vector $\boldsymbol{y}$ (pixel values of the image block to be represented), and a coefficients vector $\boldsymbol{x}$. Here, $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x}$, where $\boldsymbol{y}$

is a known column vector of size H and $\boldsymbol{x}$ is an unknown column vector of size W. Based on linear algebra, we know that this is equivalent to a set of H equations containing W unknowns. Since $W > H$ in case of over-complete dictionary, we have an infinite number of solutions. In this Section, we will focus on how K-SVD can solve this problem.

This brings us to the term – ill-posed problem, i.e., there are multiple solutions that satisfy the condition, and it is impossible to determine which one is more appropriate, so a constraint needs to be made, adding the constraint that $\boldsymbol{x}$ is as sparse as possible, that is, $\boldsymbol{x}$ has as many zeros as possible, i.e., $\text{norm}(x, 0)$ is as small as possible. We will focus on how K-SVD can solve this problem.

3.1 K-SVD Algorithm

K-SVD is an algorithm that combines the ideas of k-means and SVD. The concept of K-SVD design is described as follows in conjunction with the steps of the k-means algorithm:

1. The k samples in the sample set are randomly selected as centroids. This step can be considered as the initialization of the dictionary, i.e., k bases are randomly initialized.

2. Classify the sample into the category corresponding to the nearest centroid by calculating the distance between the sample and the centroid. This step can be considered as the initialization of the coefficient matrix. There is only one item that is not zero in each column of the coefficient matrix, and the rest items are all zero. For each sample (i.e. column vector), the index with non-zero item corresponds to its category index of the sample.

3. For the categories that have been grouped, recalculate the centroid for each sample category. This step can be considered as an optimization of the base matrix.

4. Reclassification according to the newly calculated centroids. This step can be seen as an optimization of the coefficient matrix, where the base matrix, i.e., each column vector of the dictionary, can be seen as a centroid.

5. The optimization process repeats until the distance between the centroids before and after the update is less than the specified threshold.

The K-SVD dictionary learning algorithm is shown in Figure 3.3. Concretizing the above idea to the algorithm of K-SVD dictionary learning, the dictionary $\boldsymbol{D}$ needs to be initialized, either by using a pre-prepared dictionary matrix or by randomly selecting k samples

from the training set $\boldsymbol{Y}$ as column vectors of $\boldsymbol{D}$. After that, fix the dictionary and obtain the sparse coding for each sample using Eq. (3.4).

To expand on this, let's assume that the single sample is a vector $\boldsymbol{y}$, and we already know that the dictionary $\boldsymbol{D} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3, \boldsymbol{\alpha}_4]$, the goal is to compute the coefficients $\boldsymbol{x}$ for $\boldsymbol{y}$, making $\boldsymbol{x}$ as sparse as possible.

Algorithm 1: K-SVD

Input: training sample $\boldsymbol{Y}$, dictionary $\boldsymbol{D}$(optional), coefficient matrix $\boldsymbol{X}$

Output: dictionary $\boldsymbol{D}$, coefficient matrix $\boldsymbol{X}$

Initialization: If there is no initial dictionary, take K column vectors randomly from the training sample $\boldsymbol{Y} \in \mathbf{R}^{m \times n}$ or take the first K column vectors of the left singular matrix of $\boldsymbol{Y}$ as the base vectors of the initial dictionary to obtain the dictionary $\boldsymbol{D}^{(0)} \in \mathbf{R}^{m \times K}$. Set $j = 0$.

Stop Rules: The maximum number of iterations is reached, or converges to the specified error.

while *No* ***Stop Rules*** *triggered* **do**

Sparse Coding Stage: Use OMP and $\boldsymbol{D}^{(j)}$ generated in the previous step to compute the coefficient matrix $\boldsymbol{X}$, by approximating the optimization problem:

$$\arg\min_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|^2, \ s.t. \ \|\boldsymbol{X}\|_0 \leq T_0$$

Dictionary Updating Stage: For each column in $\boldsymbol{D}^{(j)}$, update $\boldsymbol{d}_k \in \boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_k$ by:

1. Calculate the residual matrix $\boldsymbol{E}_k$, $\boldsymbol{E}_k = \boldsymbol{Y} - \sum_{j \neq k} \boldsymbol{d}_j \boldsymbol{x}_j^T$
2. Extract the index group $\omega_k = i | 1 \leq i \leq n, \boldsymbol{x}_k^T(i) \neq 0$ and non-zero items from $\boldsymbol{x}_k^T$ as $\boldsymbol{x'}_k^T = \boldsymbol{x}_k^T(i) | 1 \leq i \leq n, \boldsymbol{x}_k^T(i) \neq 0$
3. Extract columns corresponding to ω_k from $\boldsymbol{E}_k$, and obtain $\boldsymbol{E}'_k$.
4. Perform SVD on $\boldsymbol{E}'_k$. $\boldsymbol{E}'_k = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, use the first column of $\boldsymbol{U}$ to update $\boldsymbol{d}_k = \boldsymbol{U}(\cdot, 1)$, $\boldsymbol{x'}_k^T = \boldsymbol{\Sigma}(1,1)\boldsymbol{V}^T(1, \cdot)$, then replace $\boldsymbol{x}_k^T$ by $\boldsymbol{x'}_k^T$.
5. Set $j = j + 1$

Figure 3.3: K-SVD dictionary learning algorithm

Here we need to first find the base vector that is closest to the vector y by calculating the point product of $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3$, and $\boldsymbol{\alpha}_4$ with $\boldsymbol{y}$ separately, the $\boldsymbol{\alpha}$ corresponding to the maximum value of the product is the target. Assuming that the product of $\boldsymbol{\alpha}_2$ and $\boldsymbol{y}$ is maximal, we then use $\boldsymbol{\alpha}_2$ as the first base, and the initial coding vector is then $\boldsymbol{x}_1 = (0, b, 0, 0)$, where b is an unknown coefficient.

At this point the equation $\boldsymbol{y} - b * \boldsymbol{\alpha}_2 = 0$ has only one unknown parameter b, which is equivalent to solving the least squares problem. Then reconstruct the data by multiplying b with $\boldsymbol{\alpha}_2$ and compute the residual vector $\boldsymbol{y}' = \boldsymbol{y} - b * \boldsymbol{\alpha}_2$. If the residual vector $\boldsymbol{y}'$ satisfies the reconstruction error threshold range ε, then the operation is over, otherwise go to the next step.

From the dictionary, find the nearest vectors of the remaining base vectors $\boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_3$, and $\boldsymbol{\alpha}_4$ to the residual vector $\boldsymbol{y}'$, assuming that $\boldsymbol{\alpha}_3$ is the target vector, and then make the new coding vector: $\boldsymbol{x}_2 = (0, b, c, 0)$, where b and c are unknown coefficients. This leads to the equation: $\boldsymbol{y} - b * \boldsymbol{\alpha}_2 - c * \boldsymbol{\alpha}_3 = 0$. There are two unknown coefficients b and c in the equation, which can be solved here using the OMP algorithm[19]. Update the residual vector $\boldsymbol{y}' = \boldsymbol{y} - b * \boldsymbol{\alpha}_2 - c * \boldsymbol{\alpha}_3$ again, if $\boldsymbol{y}'$ satisfies the threshold range then it's over, otherwise the loop keeps going. Expanded description ends here.

At this point, we know the coding of the sample, i.e., the coefficient matrix. The next goal is to update the dictionary as well as the coefficients. K-SVD uses a column-by-column method to update the dictionary, i.e., when the kth base vector is updating, the other base vectors are fixed. Suppose we currently want to update the kth base vector $\boldsymbol{\alpha}_k$ so that the kth row vector corresponding to the coefficient matrix $\boldsymbol{X}$ is $\boldsymbol{x}_k$, then the objective function is as follows:

$$\begin{aligned} \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|^2 =& \|\boldsymbol{Y} - \sum_{j=1}^{K} \boldsymbol{d}_j \boldsymbol{x}_j^T\|^2 \\ =& \| \left(\boldsymbol{Y} - \sum_{j \neq k} \boldsymbol{d}_j \boldsymbol{x}_j^T \right) - \boldsymbol{d}_k \boldsymbol{x}_k^T \|^2 \\ =& \|\boldsymbol{E}_k - \boldsymbol{d}_k \boldsymbol{x}_k^T\|^2 \end{aligned} \tag{3.5}$$

Where $\boldsymbol{d}_k$ is the kth column vector of $\boldsymbol{D}$, while $\boldsymbol{x}_k^T$ is the kth row vector of the coefficient matrix $\boldsymbol{X}$. Thus, the optimization problem can be described as:

$$\min_{\boldsymbol{d}_k \boldsymbol{x}_k^T} \|\boldsymbol{E}_k' - \boldsymbol{d}_k \boldsymbol{x'}_k^T\|^2 \tag{3.6}$$

Here, we perform SVD on $\boldsymbol{E}_k'$.

$$\boldsymbol{E}'_k = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{3.7}$$

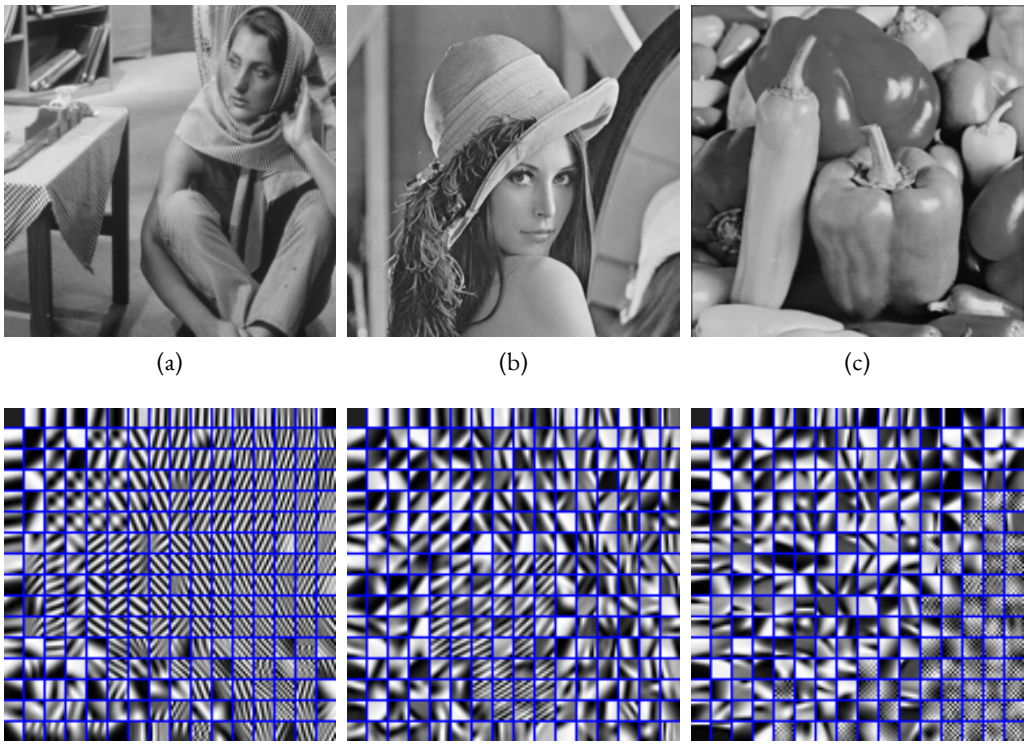
When the singular values in the matrix $\boldsymbol{\Sigma}$ are arranged in descending order, the 1st column vector $\boldsymbol{u}_1 = \boldsymbol{U}(\cdot, 1)$ of the matrix $\boldsymbol{U}$ is taken as $\boldsymbol{d}_k$, i.e., $\boldsymbol{d}_k = \boldsymbol{u}_1$, and the product of the 1st row vector of the matrix $\boldsymbol{V}^T$ and the 1st singular value is taken as $\boldsymbol{x}'^T_k$, i.e.,$\boldsymbol{x}'^T_k = \boldsymbol{\Sigma}(1,1)\boldsymbol{V}^T(1,\cdot)$. After obtaining $\boldsymbol{x}'^T_k$, update it correspondingly to the original $\boldsymbol{x}^T_k$.

Updating the dictionary column by column will generate a new dictionary, and the K-SVD algorithm iteratively updates the coefficient matrix and the dictionary until convergence.

3.2 Performance of Sparse Representation

When an image is expressed under the condition that the number of non-zero coefficient is T_0 or less, the dictionary designed by K-SVD under the constraint that the number of non-zero coefficients is less than T_0 using that image can minimize reconstruction errors. However, the characteristics of K-SVD derived dictionaries are highly dependent on the feature of the images used in training. A dictionary trained for a specific image is optimum for that image, but not necessarily for other images.

Figure 3.4 shows the dictionaries $D_{barbara}$, D_{lena} and D_{pepper} designed by K-SVD under the constraint condition of $T_0 = 3$ for each of the three images "Barbara", "Lena" and "Pepper". The atoms included in each dictionary reflect the characteristics of the image used for training. For example, in the Barbara image, stripe patterns having various directions present in the original image strongly appear as several atoms in the dictionary. To measure the sparse representation performance of these dictionaries, the Peak Signal to Noise Ratio (PSNR) values of the images reconstructed by applying the three dictionaries $D_{barbara}$, D_{lena} and D_{pepper} to each image, "Barbara", "Lena" and "Pepper" are shown in Table 3.1. Table 3.1 shows the results under the same sparsity condition of $T_0 = 3$. The quantization and entropy coding for coefficients are not applied because they affect the evaluation of sparse representation performance itself. Therefore, note that the PSNR in Table 3.1 indicates the sparse representation performance, not coding performance. That is, first, three non-zero coefficients are obtained by repeating the projection (OMP) of the target block vector in the image onto the basis in each dictionary three times. After that, the target image is reconstructed using only those three coefficients, and how much it can approximate the original image was measured. For comparison, the result when each image is



(a) (b) (c)

(d) (e) (f)

Figure 3.4: Dictionaries designed by K-SVD

The original images ((a), (b), (c)) and their corresponding dictionaries ((d), (e), (f))

reconstructed using the HEVC-DCT[44] dictionary under the same sparsity condition of $T_0 = 3$ is also shown. For all images, image reconstruction performance is highest with the dictionary trained against itself. Unfortunately, using a dictionary trained against different images seriously degrades image reconstruction performance.

For example, for the Barbara image, the other two K-SVD dictionaries yield worse performance than HEVC-DCT. When applying K-SVD to image coding, the decoder has to use the same dictionary as the encoder, so the designed dictionary itself must be encoded and transmitted to the decoder. However, the coding and transmission of atoms for each image every time incurs large overheads for information transmission, and is not practical from the viewpoint of rate distortion performance.

Table 3.1: PSNR values of the images reconstructed by the different dictionaries

	$D_{barbara}$	D_{lena}	D_{pepper}	$D_{HEVC-DCT}$
Barbara	**29.59**	25.53	25.08	26.20
Lena	31.70	**32.95**	32.27	29.51
Peppers	31.70	32.31	**32.94**	29.67

3.3 Single-Class Dictionary Learned from Multiple Images

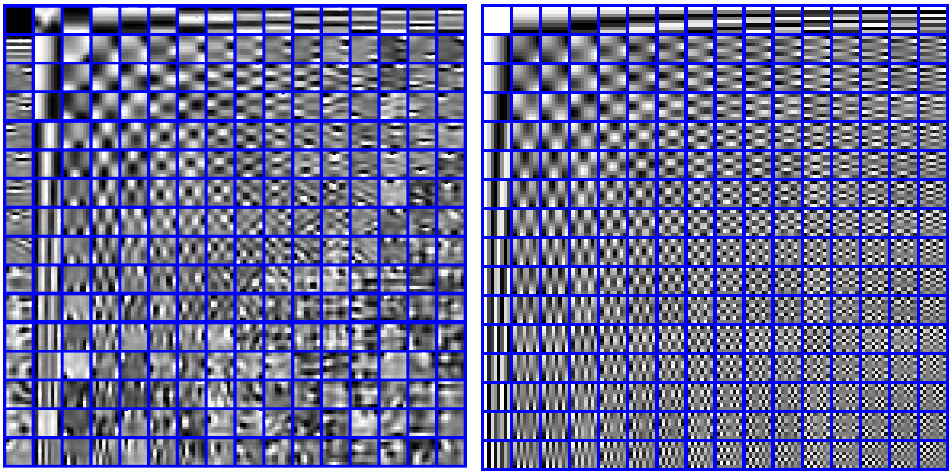
When it comes to the use of learning dictionaries for image compression, the easiest way to think of it is to learn from the original image and use the obtained dictionary for image compression. As mentioned above, learned dictionary can achieve better compression efficiency than traditional DCT in terms of sparse representation, but it is obvious that different images generate different dictionaries, so the transmission needs to take into account the additional information generated by the update of the dictionary, that is, the information entropy of the transformation coefficients will be reduced while the entropy of the additional information will increase, which, overall, will increase the transmission bit rate.

Another approach is to use big data, using a large number of images for learning, in order to reach an improvement in the general purpose performance of the learned dictionary. Given the computational complexity of dictionary learning algorithms, it is common practice to split the image into blocks of the same size such as 8×8 etc. as in existing standards such as JPEG, and use a large number of blocks for dictionary learning.

40 full-HD (1920×1080) images are used and decomposed into 8×8 sized blocks, in other words, using a total of about 1.3 million blocks, performed dictionary learning using K-SVD, and the over-complete dictionary obtained with the sparsity constraint parameter $T_0 = 5$ is shown in Figure 3.5. Comparing the over-complete DCT dictionaries with the same size, we found a high degree of similarity between the two.

This proves on the one hand that DCT can guarantee high compression efficiency for different kinds of samples, and on the other hand, it is expected that as the training data increases, the dictionaries learned in this way will be more similar to the DCT dictionaries, i.e., the advantage of using learning dictionaries - to get an approximately optimal transform dictionary for different images - will be lost.

In order to avoid this problem, we attempted to perform dictionary learning by pre-grouping the training samples according to their respective local characteristics and per-



(a) Dictionary learned by K-SVD (b) DCT

Figure 3.5: Comparison between learned dictionary and DCT

forming dictionary learning on the different groups separately, which will be described in detail in the following chapter.

3.4 Summary

In this chapter, we described the design ideas and computational steps of the K-SVD algorithm in section 3.1. After that, in section 3.2, three images were used to cross-test the performance of the sparse representation. Using the three images' own data as training samples, the corresponding three dictionaries are generated, additional with the DCT, for a total of four dictionaries. The performance of these four dictionaries are measured on each image, and the results proved that the dictionary generated from its own data has the best performance for the corresponding image, however, for the other two images, its performance is not satisfactory. Following in section 3.3, how learning dictionaries generated by algorithms such as K-SVD can be applied to image compression applications is discussed, and it was clarified that customizing dictionary design for each transmitted image is difficult to apply in practice, while it was found that single-class dictionary generated from big data is highly similar to DCT dictionaries in terms of visual performance. This means that such learned dictionary is difficult to produce a breakthrough in compression efficiency compared to the DCT dictionary.

Dictionary Design based on Multi-class K-SVD with Iterative Class Update

4

Contents

K-SVD is a popular technique for learning a dictionary that offers sparse representation of the input data. Given an image signal, K-SVD can derive a dictionary that well approximates each block with a sparse combination of atoms from the set of blocks composing the image[18]. An example of this approach is the facial image codec based on the K-SVD dictionary introduced by reference[17].

It is known that dictionaries generated by K-SVD are largely dependent on the features of the training images. As mentioned in Chapter 3, when applying K-SVD to image coding, the decoder has to use the same dictionary as the encoder, so the designed dictionary itself must be encoded and transmitted to the decoder. However, the coding and transmission of atoms for each image every time incurs large overheads for information transmission, and is not practical from the viewpoint of rate distortion performance.

Therefore, the extension of K-SVD to support multiple dictionaries is a promising approach to more efficient representations of natural images with various features. Here, let us call the extended K-SVD "multi-class K-SVD". Multi-class K-SVD adaptively selects the most suitable dictionary based on the local feature(s) of the image to be encoded.

4.1 Multi-class Dictionary without Class Update

4.1.1 Related work

In order to design multi-class dictionary, some researchers take approaches that share the multiple dictionaries among an encoder and a decoder in advance. That is, first, the image is divided into small blocks to calculate local features, then a set of blocks having similar features is created as a class, and finally, K-SVD is executed for each class so as to create multiple dictionaries.

It is considered that the local features in images can be classified into similar geometric patterns such as the direction of edges and texture features. A set of dictionaries designed for each class is shared in advance by the encoder and decoder, and these are adaptively switched when encoding. This eliminates the need to encode new sets of dictionary information, and makes it possible to represent more images efficiently.

For example, in reference[45] , based on the H.265/HEVC framework, an approach is proposed in which training samples are classified by transform unit (TU) size and a dictionary is designed by K-SVD for each class. Moreover, it is described that an application-specific dictionary can be developed for particular applications such as gaming or medical video by changing samples in training.

Further, in reference[46], block classification based on intra-frame/inter-frame prediction residual power is performed using the H.264/AVC framework, and different dictionaries are designed class by class. For application other than image coding, multiple dictionaries designs based on K-SVD also have been studied.

In reference[47] , multiple dictionaries are designed in order to sharpen the character image. Small patches in images are classified into 13 classes based on their pixel distribution state, and a K-SVD dictionary is designed for each class.

In reference[48], the effectiveness of class specific sparse codes is investigated in the context of discriminative action classification. The local motion features for each action are trained by K-SVD to design the action specific dictionary.

4.1.2 Multi-class dictionary for image coding

An attempt has also been made to generate a multi-class sparse dictionary based on local features of images, aiming to apply it to image coding[49]. As shown in Figure 4.1 a large amount of image blocks are classified into multiple classes based on their local features.

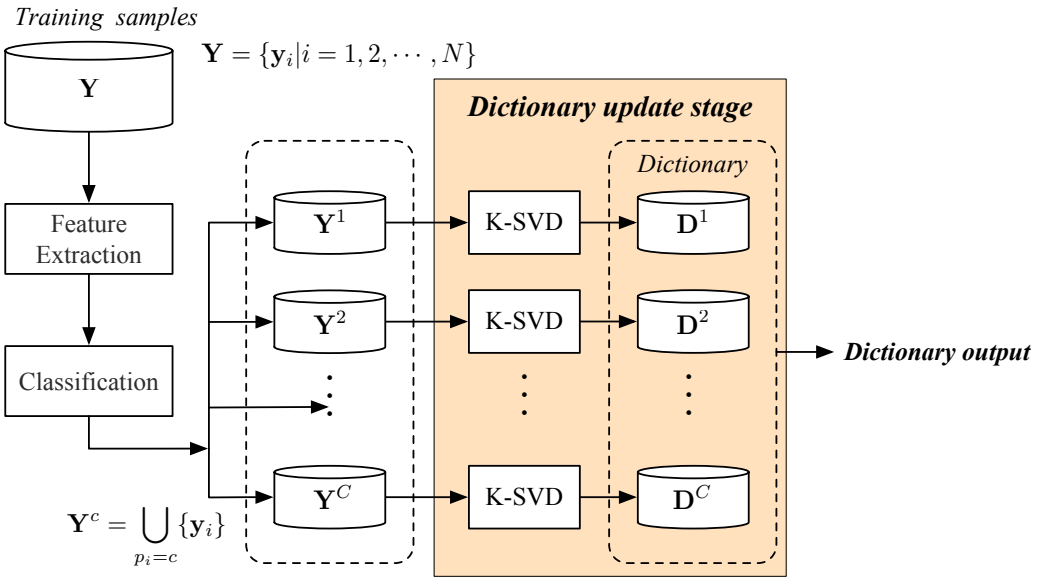


Figure 4.1: Block diagram of Multi-class K-SVD dictionary design

Scale-invariant feature transform (SIFT)[50] is well known as a robust algorithm to detect and describe local features in images. In SIFT, points lying on high-contrast regions such as object edges or corners are chosen as the key points, as their features are invariant to any scaling, rotation or translation of the image. However, for applications such as object recognition and image coding, not only features of the well-detectable points but also those of points on gradation or flat regions are still important. In other words, it is desirable to design different bases set for points having same SIFT feature under different scale, orientation and brightness. Therefore, Dense Scale Invariant Feature Transform (DSIFT)[51] is hired instead of SIFT. In DSIFT, features are calculated under fixed points and fixed orientation. Feature descriptor of DSIFT is similar to SIFT, that is, it is a spatial histogram of the image gradient. In the neighborhood region around each key point, the spatial coordinates are quantized into four each (4×4 sub-regions), and orientations of gradient are quantized into eight bins. Then, a histogram consisting of $8 \times 4 \times 4 = 128$ bins is produced for each key point.

After the local features extracted, the large number of image blocks are classified into multiple classes by k-means clustering[52] based on their local features. Then, K-SVD is adopted for dictionary training. Here, a dictionary which contains multiple sparse bases is learned.

4.2 Proposal of Dictionary Design Method Using Class Update

It has been clarified that multi-class K-SVD gives better coding performance than single class K-SVD (i.e. with one dictionary). In references[45–49] local regions of an image are classified into multiple classes according to their characteristics, and a dictionary setting based on K-SVD is performed class by class. However, conventional methods do not consider the effects of classification on the subsequent processes, a dictionary learning based on K-SVD, because these methods use pre-determined features in performing classification. This is, the classifications and dictionary learning are designed independently. This raises the disadvantage that image representation performance depends on what kinds of features are used for classification. Therefore, there still remains the potential for improvements in coding efficiency by combining dictionary training and classification optimization.

Here, a propose of a multi-class K-SVD method that takes into account the interplay between classification and dictionary design. The proposed method iterates the classification of training samples and dictionary design for each class. In order to focus on application to image coding, its classification takes an unsupervised approach based on the k-means method, unlike the conventional supervised technique for such as object recognition and image classification. In the proposed method, a multi-class dictionary is designed by repeating the two stages, unsupervised class update stage for all training vectors and dictionary update stage for each class by K-SVD. Compared to conventional approaches that do not perform classification optimization, better compression efficiency is expected.

4.2.1 Algorithm

In this section, I propose a new multi-class dictionary design method consisting of two stages (dictionary design and classification), and its application for image coding. For designing multi-class dictionaries, the classification of training data and the dictionary design process for each class are iterated. By performing iterative convergence processing, we can create multi-class dictionaries that solve the conventional problem of what kind of local features should be used for clustering. In the proposed method in this dissertation, dictionary design including classification of training data is performed as unsupervised learning. Therefore, the method includes a different process from the conventional dictionary design method based on iterative updating using supervised data. In this section, a detailed algorithm to design the multiple dictionaries based on K-SVD under unsupervised training is shown, and how the multi-class dictionaries obtained by the proposed method can be applied to

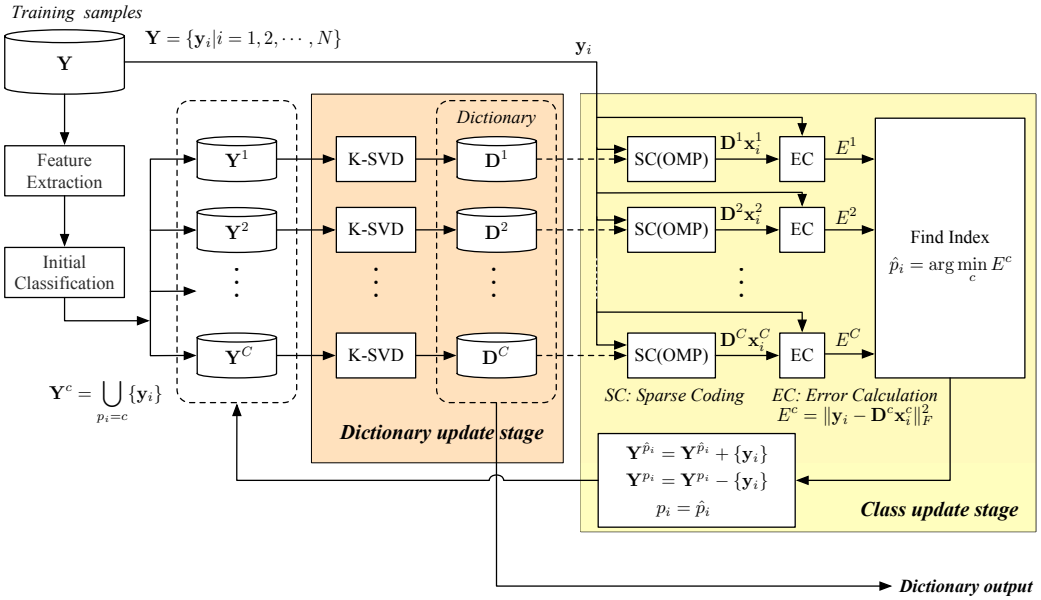


Figure 4.2: Block diagram of Multi-class K-SVD dictionary design with iterative class update

image compression.

Figure 4.2 depicts the process flow of the proposed multi-class K-SVD dictionary design method. Since dictionary design and classification cannot be optimized at the same time, the proposal alternately performs, for each class, the classification of training data and dictionary design. Figure 4.3 details the steps of the multi-class dictionary design; the specific procedures in each stage are described as follows.

First, training images are divided into small blocks, and training vectors are calculated to yield a set of m-dimensional vectors $\boldsymbol{y}_i (i = 1, 2, ..., N)$ where elements of $\boldsymbol{y}_i$ are pixel values in the i-th block. N is the number of training vectors. We consider that each training vector, $\boldsymbol{y}_i$, can be approximated as a weighted linear combination of the atoms in dictionary $\boldsymbol{D}^c$, where the weight coefficients of the atoms are denoted as coefficient vector $\boldsymbol{x}_i$. Training vectors are initially classified into C classes, $\boldsymbol{Y}^1, \boldsymbol{Y}^2, \cdots, \boldsymbol{Y}^C$, based on their local features and class index for $\boldsymbol{y}_i$ is determined as p_i, where $p_i \in 1, 2, 3, \ldots, C$. As the local feature, dense scale invariant feature transform (DSIFT)[51] is utilized because the distribution of edge gradient direction in a block can be expected to significantly influence the shape of basis patches to be designed. The initial dictionary for each class is set to over-complete DCT with size K.

Algorithm 2: Multi-class K-SVD

Initialization:
Set the number of classes C.
Set the maximum number of iterations for class update $L_{\max}$.
Set over-complete DCT to initial $\boldsymbol{D}^c$ for each class.
Classify all training vectors $\boldsymbol{y}_i (i = 1, 2, \cdots, N)$ into classes based on DSIFT, $\boldsymbol{Y}^C = \bigcup_{p_i=c} \{\boldsymbol{y}_i\}$.
Set class index for $\boldsymbol{y}_i$ to p_i.
Set $count = 0$.
Dictionary Updating Stage:
for $c = 1, 2, \cdots, C$ **do**
  Calculate dictionary $\boldsymbol{D}^c$ with K-SVD.
Class Update Stage:
for $i = 1, 2, \cdots, N$ **do**
  for $c = 1, 2, \cdots, C$ **do**
    Find $\boldsymbol{x}_i^c$ that approximates $\boldsymbol{y}_i$ by sparse coding with $\boldsymbol{D}^c$.
  New class index $\hat{p}_i = \arg\min_c \|\boldsymbol{y}i - \boldsymbol{D}^c \boldsymbol{x}_i^c\|_F^2$.
  if $\hat{p}_i \neq p_i$ **then**
    $\boldsymbol{Y}^{\hat{p}_i} = \boldsymbol{Y}^{\hat{p}_i} + \boldsymbol{y}_i$
    $\boldsymbol{Y}^{p_i} = \boldsymbol{Y}^{p_i} - \boldsymbol{y}_i$
    $p_i = \hat{p}_i$
Set $count = count + 1$.
Convergence Check:
if $count > L_{\max}$ *or no class index is changed* **then**
  Output: Dictionary $\boldsymbol{D}^c (c = 1, 2, \cdots, C)$
else
  Back to "**Dictionary Update Stage**"

Figure 4.3: Multi-class K-SVD dictionary design algorithm

Next, in the dictionary update stage, a dictionary that enables sparse representation for training vectors for each class is designed. In preparation, I concatenate all training vectors $\boldsymbol{y}_i$ that belong to class c as columns of matrix $\boldsymbol{Y}^c \in \boldsymbol{R}^{m \times n(c)}$ and similarly concatenate coefficient vectors $\boldsymbol{x}_i$ for $\boldsymbol{y}_i$ to build matrix $\boldsymbol{X}^c \in \boldsymbol{R}^{K \times n(c)}$. Here, $n(c)$ is the number of training vectors belonging to the c-th class. Dictionary $\boldsymbol{D}^c$ and coefficient $\boldsymbol{X}^c$ are obtained by solving Eq.(4.1).

$$\min_{\boldsymbol{X}^C, \boldsymbol{D}^C} \|\boldsymbol{Y}^C - \boldsymbol{D}^C \boldsymbol{X}^C\|_F^2, \ s.t. \ \forall_i, \|\boldsymbol{x}_i^C\|_0 \leq T_0 \tag{4.1}$$

As the solver of the problem in Eq.(4.1), I use K- SVD. Specifically, as shown in Eq.(4.1), dictionary $\boldsymbol{D}^C \in \mathbf{R}^{m \times K}$ and coefficients $\boldsymbol{X}^c$ that minimize the square error between original signal $\boldsymbol{Y}^C$ and its reconstructed signal $\boldsymbol{D}^c\boldsymbol{X}^c$ are found under the sparsity constraint that the number of nonzero coefficients in each column of matrix $\boldsymbol{X}^c$ is equal to or less than T_0. This designs C dictionaries.

After dictionary design, the class update stage is executed. Among the C kinds of dictionaries $\boldsymbol{D}^c$, a new class index $\hat{p}$ for each training vector $\boldsymbol{y}_i$ is re-assigned so as to satisfy the following:

$$\hat{p} = \arg\min_c \|\boldsymbol{y}_i - \boldsymbol{D}^c\boldsymbol{x}_i^c\|_F^2, \ s.t. \ \|\boldsymbol{x}_i^c\|0 \leq T_0, i = 1, 2, \cdots, N \qquad (4.2)$$

The above equation yields the class index that minimizes the square error between a training vector and its reconstructed vector from the designed dictionary subject to the constraint on the number of nonzero coefficients. If $\hat{p}_i$ is different from p_i, the training sample $\boldsymbol{y}_i$ is moved to class $\boldsymbol{Y}^{\hat{p}_i}$ and the class index p_i of $\boldsymbol{y}_i$ is replaced by $\hat{p}_i$.

These two steps, the dictionary update stage for each class by K-SVD and the class update stage, are iterated until the convergence conditions are satisfied. The convergence conditions are that the number of iterations exceeds predetermined threshold $L_{\max}$ or that none of the class indices of the training vectors change. Finally, the algorithm outputs $\boldsymbol{D}^c(c = 1, 2, \cdots, C)$ as the multi-class dictionaries.

In order to effectively represent the actual image, each dictionary to be designed should contain one DC basis[18, 53], as has been confirmed. Therefore, one DC basis is included in the initial dictionary for each class, and the DC basis is not changed during iterative processing. Also, atoms other than DC maintain zero mean during iterative processing.

4.3 Application to Image Coding

This section describes the application to image coding process (encoding / decoding) using dictionaries designed by the proposed multi-class K-SVD algorithm.

A block diagram of the encoder and decoder is shown in Figure 4.4. All dictionaries $\boldsymbol{D}^c(c = 1, 2, \cdots, C)$ are designed offline, and prestored in both encoder and decoder. Here, all atoms other than DC in the dictionary are normalized so that the mean value is zero and the standard deviation is one.

In the encoding process, an image to be coded is divided into small blocks of the same size as the used in the training process. Then, OMP is performed for each target block $\boldsymbol{t}_i$

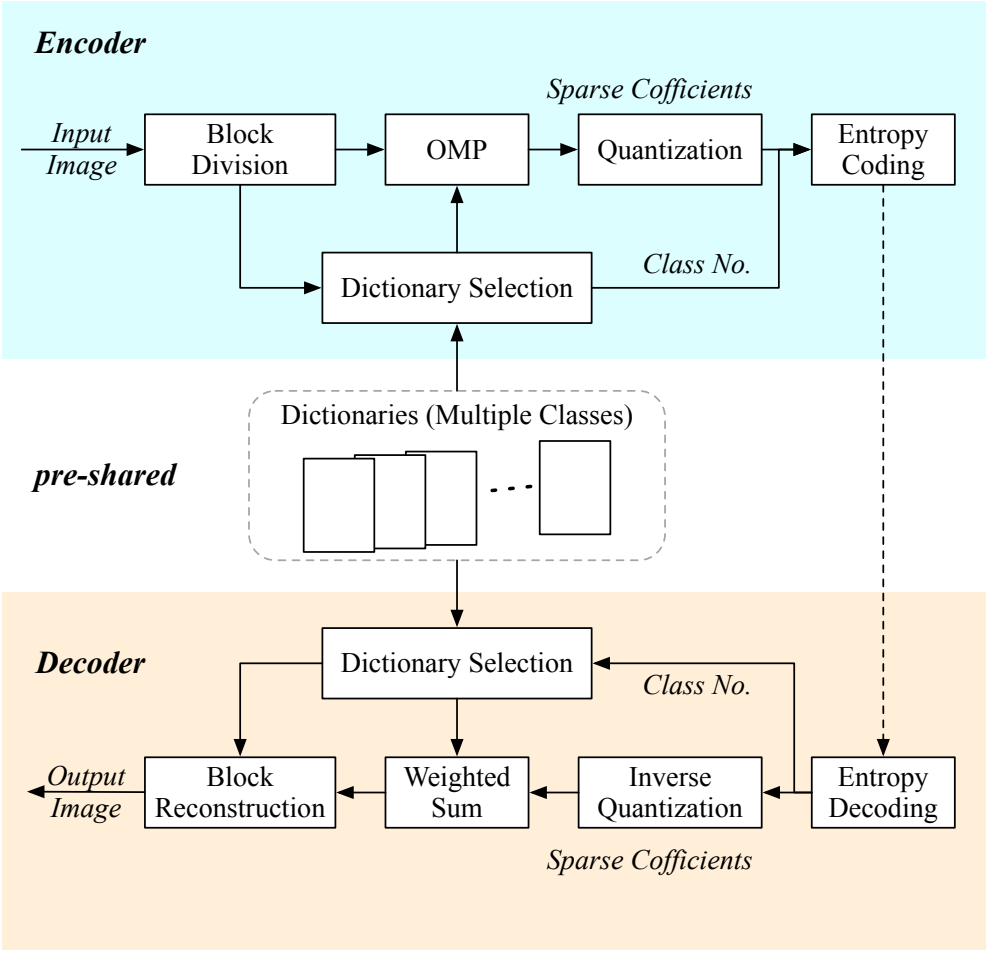


Figure 4.4: Block diagram of encoder and decoder with multi-class dictionaries

under sparsity condition T_0; the squared errors $\boldsymbol{e}^c(c = 1, 2, \cdots, C)$ are calculated as follows,

$$\boldsymbol{e}^c = \|\boldsymbol{t}_i - \boldsymbol{D}^c \boldsymbol{x}_i\|_F^2 \tag{4.3}$$

then the class index c and sparse coefficients $\boldsymbol{x}_i$ that minimize squared error $\boldsymbol{e}^c$ are determined. Quantized coefficients $Q(\boldsymbol{x}_i)$ and class index c are encoded and transmitted.

When sparse coding is applied to image compression, how to assign codes to sparse coefficients is an important point. Due to the nature of sparse coding, most of the coefficients for representing image data are zero; at most T_0 coefficients are non-zero. In order to reconstruct an image from sparse coefficients, it is necessary to efficiently encode the index of each nonzero coefficient, which indicates the basis corresponding to the nonzero coefficient, based on its statistical properties. Previous studies have shown that the index at which

nonzero coefficients occur is uniformly random, and the distribution of the quantization level of nonzero coefficients can be approximated by a Laplacian function[45]. In reference[45], based on this characteristic, indices of the nonzero coefficients are represented by a fixed length code and the quantized level of the nonzero coefficients is represented by a Golomb-Rice code. To express the indices of nonzero coefficients more efficiently, a method of assigning a variable length code to a zero-run length (i.e. the number of consecutive zero coefficients between nonzero coefficients) has been proposed[1]. The entropy coding in this paper follows the method of reference[1]. That is, for each block, the number of quantized nonzero coefficients (NUM), the zero-run length between nonzero coefficients (ZR), and the level number of the quantized nonzero coefficients (LEVEL) are separately encoded and transmitted. Since the DC coefficient of a block is highly correlated with that of its prior block, differential pulse code modulation (DPCM) is adopted when coding DC coefficients.

Furthermore, in the proposed method, it is necessary to encode the class index in order to identify the dictionary used for sparse coding. According to our experiments, there was no clear correlation between the class index of a target block and the class index of its neighboring blocks. Moreover, no indication of the occurrence probability distribution of the class index concentrated on a specific class. Therefore, class indices are represented using $\lceil \log_2 C \rceil$ bit fixed length codes.

In the decoding process, the dictionary is adaptively selected block by block based on the decoded class index, and pixel values in the block are reconstructed as the sum of atoms weighted by the decoded sparse coefficients. Since the weighted sum is only calculated for the inverse transformation, the computation cost in the decoder is as low as a normal DCT.

4.4 Experimental Results

4.4.1 Simulation conditions

The effectiveness of the proposed method is evaluated using the ITE test image data set[54, 55] and HEVC test sequence[44]. The experimental conditions are summarized in Table 4.1. Thirty-one images with a resolution of 1080p are used as training data for dictionary design, and six images in Figure 4.5, not included in the training images, are used as encoding targets.

In this experiment, in order to analyze the behavior of the proposed method in detail, multi-class dictionaries are designed using various parameters. Specifically, dictionaries

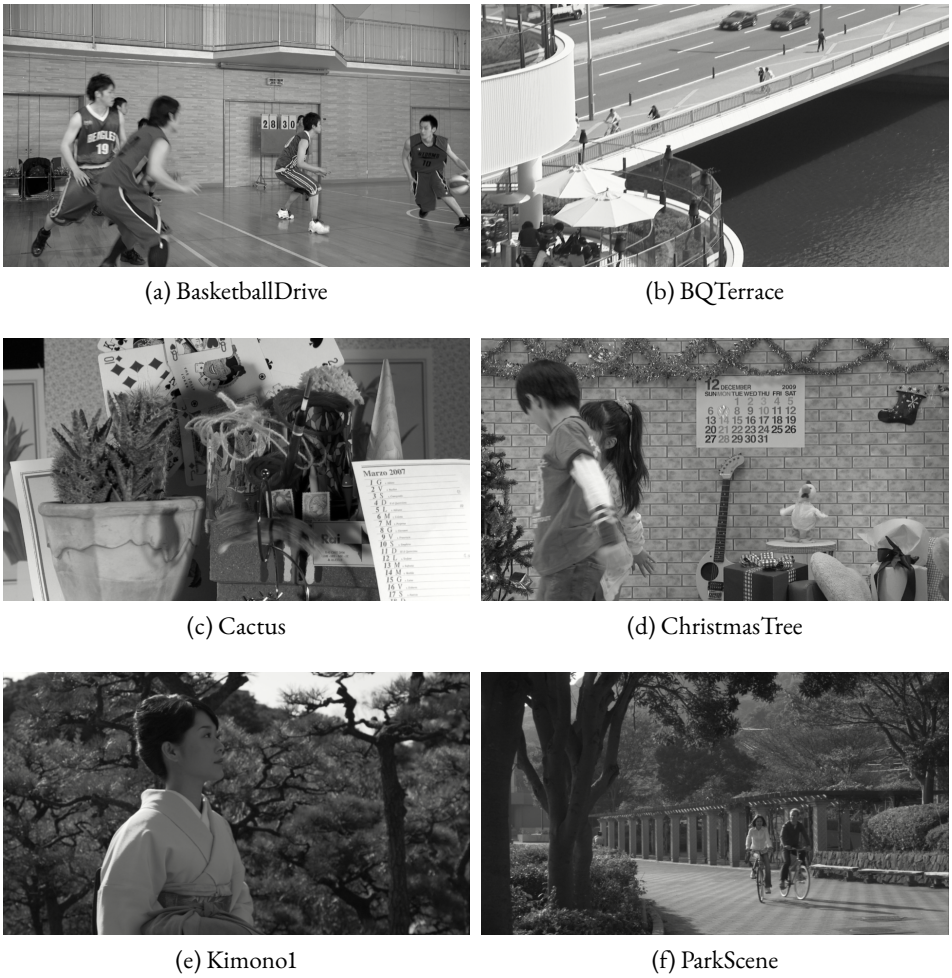
Table 4.1: Experimental conditions

Dictionary Training	Block size	8×8
	Initial dictionary	Over-complete DCT with 16×16 atoms
	Number of classes C	32, 64, 128
	Sparsity parameter T_0	3, 5, 7, 9
	Initial classifier	k-means using DSIFT
	Maximum iteration limit for class update, $L_{\max}$	20
Encoding	Coefficients quantization	Uniform quantization for step QP (for DC, QP=1)
	Coefficients coding (DC)	DPCM using the previous block's DC coefficient
	Coefficients coding (AC)	Separately encoding of NUM, ZR, and LEVEL
	Class index	Fixed length coding

are prepared by setting the class index C to $32, 64, 128$ and the sparsity parameter T_0 to $3, 5, 7, 9$. Each basis included in the dictionary is set to 8×8 size, following JPEG. Dictionary training based on multi-class K-SVD is started for each class by setting overcomplete DCT with $16 \times 16 = 256$ atoms as an initial dictionary. For the initial classification, the DSIFT feature obtained for each block is used, and the maximum number of iterations of class update is set to 20.

Next, experimental conditions of encoding are as follows. The coding experiment examined intraframe coding, and performance is evaluated from the viewpoint of the PSNR of the decoded image and the amount of information transferred. After the image to be coded is divided into 8×8 blocks, sparse coding using a multi-class dictionary is performed for each block according to the method described in Section 3.2, and a class index for specifying the dictionary to be used and the coefficients corresponding to each basis are obtained. As described in Section 3.2, DC coefficients of two neighboring blocks are strongly correlated, so the difference from the decoded DC coefficient of the previous block is quantized and encoded. Since the DC coefficient greatly influences visual image quality, the quantization step for the DC coefficient difference is set to just one. The amount of information for DC coefficients is calculated as the entropy of the quantization level number corresponding to the quantized prediction error. AC coefficients (coefficients for atoms other than DC) are linearly quantized with quantization width QP. The quantization level number "LEVEL" for coefficient x is calculated as:

$$\text{LEVEL} = sign(x) \times \lfloor (|x| + \text{QP}/2) / \text{QP} \rfloor \tag{4.4}$$



(a) BasketballDrive (b) BQTerrace

(c) Cactus (d) ChristmasTree

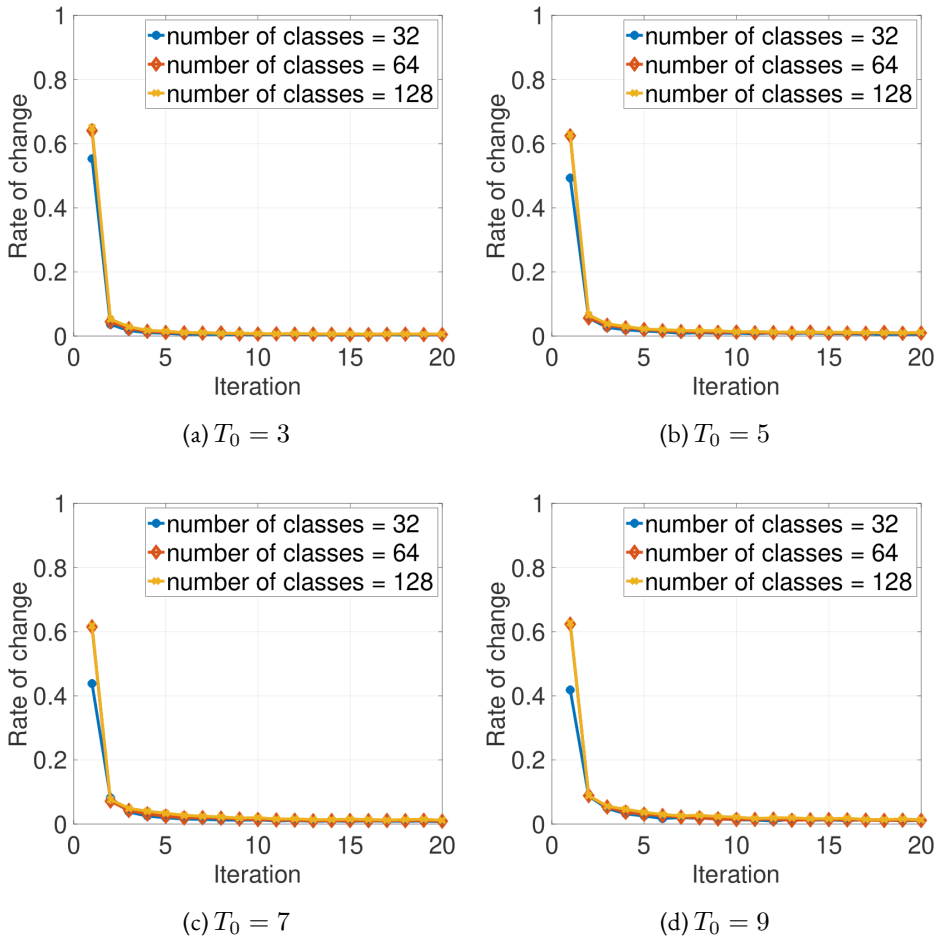
(e) Kimono1 (f) ParkScene

Figure 4.5: Test images

The amount of information for AC coefficients is calculated as the entropy obtained from the occurrence probability NUM, ZR and LEVEL. Fixed length code of $\lceil \log_2 C \rceil$ is allocated to each class index. The total amount of information required for each block is calculated as the sum of the amount of information occupied by DC coefficient, AC coefficients, and class index.

4.4.2 Dictionary training performance

Figure 4.6 shows the percentage of training samples whose class indices changed among the training samples with each class update iteration. Clearly, the change in the number



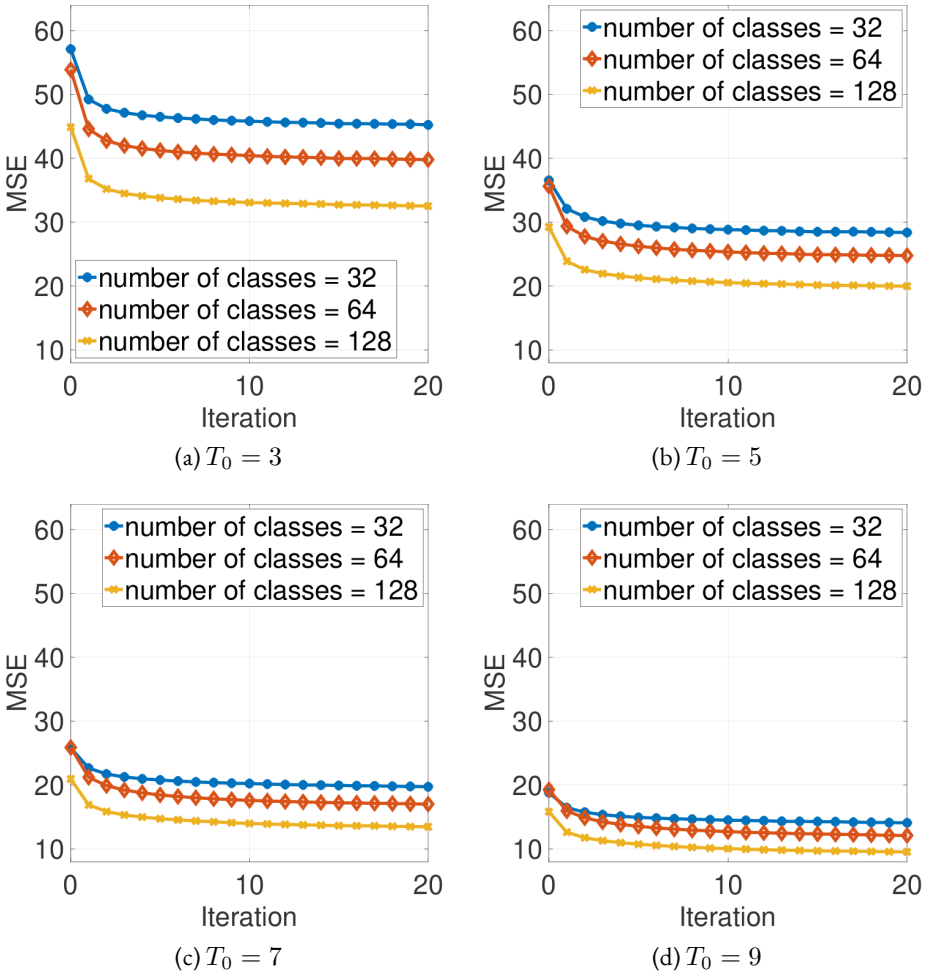
(a) $T_0 = 3$ (b) $T_0 = 5$

(c) $T_0 = 7$ (d) $T_0 = 9$

Figure 4.6: The rate of change in the number of training samples belonging to each class

of training samples belonging to each class gradually converges to zero with class update iteration regardless of the number of classes, C, or the sparsity parameter, T_0.

In addition, Figure 4.7 shows the mean square error (MSE) between the original block and the block reconstructed by the dictionary designed after each iteration number. In this determination, coefficient quantization was not performed. From Figure 4.7, we find that MSE strongly decreases in the first few iterations, continues to gradually decrease subsequent class update iterations, and converges to a basically constant value in 15 to 20 iterations regardless of C or T_0. Furthermore, the converged value of MSE apparently decreases as class number C increases and as sparsity parameter T_0 increases. This is because the atoms that offer better approximation of the local pixel value distribution are easier to find in the designed dictionaries as C and/or T_0 increase. Based on the above results, the following



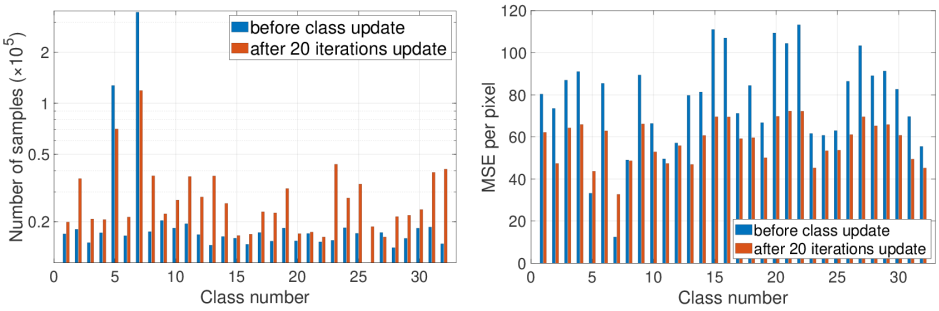
(a) $T_0 = 3$ (b) $T_0 = 5$

(c) $T_0 = 7$ (d) $T_0 = 9$

Figure 4.7: MSE convergence characteristics for the training data

experiments were carried out by setting the maximum iteration number for class update, $L_{\max}$, to 20.

Figure 4.8 shows the number of training samples belonging to each class and the MSE obtained by sparsely approximating those training samples using $T_0 = 3$ and $C = 32$. The blue bar in Figure 4.8 shows MSE of initial classification and the red bar shows MSE after 20 iterations. In the initial classification based on DSIFT, the training sample distribution concentrates on some classes, but training samples are distributed among many classes with each class update iteration. For classes whose initial classification have large MSEs, it is found that after 20 class updates the MSE decreases dramatically. It is also found that sparse approximation yields an inverse correlation between the number of samples belong-



(a) Number of the training samples

(b) MSE (per pixel)

Figure 4.8: The number of training samples in each class and their sparse representation MSE

ing to each class and MSE. This is because among the training image data, as the ratio of the low frequency vector rises above that of the high frequency vector, and the vector frequency falls, the reconstruction error becomes smaller. This tendency was also found in experiments using other numbers of classes or other sparsity parameters.

4.4.3 Initial classifier

DSIFT is based on an edge gradient histogram over orientation bins, and the atoms in the designed dictionary of each class can reflect the edge shape feature of the original block. In this subsection, we investigate how the convergence value and the convergence speed differ depending on the initial classification in the proposed algorithm. Two classification methods, block-variance-based method (VAR) and random-assignment-based method (RAND), other than DSIFT were tested. Block variance is a measure of the spread of pixel values in a block, reflecting the sharpness and complexity of the edge. It is calculated by the following equation:

$$\mathrm{VAR} = \sum_i \sum_j \left(f(i,j) - M\right)^2 \tag{4.5}$$

where M is the average of pixel values in the block. For the initial classification, the blocks are classified into C classes by k-means method based on VAR. RAND is a method to assign C random variables of 1 to C as class numbers for each block, and it does not require any specific feature calculations. The experiment by RAND is intended to verify how the proposed method behaves when starting from random initial classification.

Three types of multi-class dictionaries are designed under three initial classifications, DSIFT, VAR, and RAND. Figure 4.9 shows the convergence characteristics of MSE when

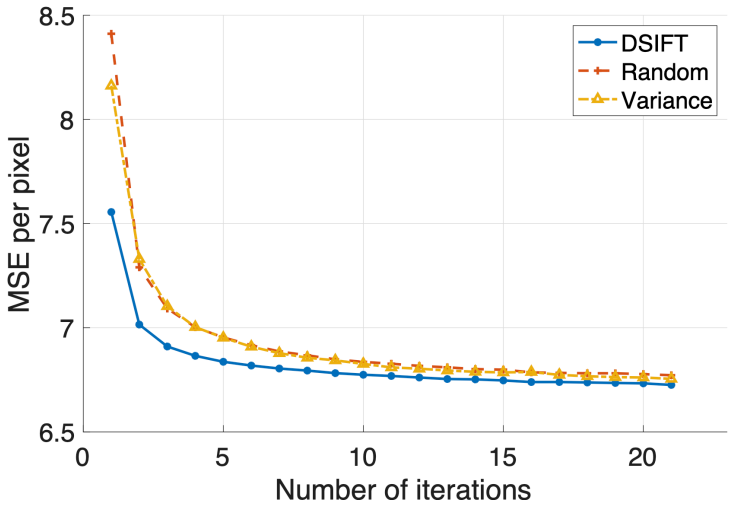


Figure 4.9: Convergence characteristics based on three kinds of initial classification methods

$C = 32$ and $T_0 = 5$. From Figure 4.9, it can be seen that the dictionaries designed by initial classification with RAND without class update (i.e. the first K-SVD output dictionaries) have poor image representation performance. Moreover, MSE by the dictionaries designed by the initial classification by DSIFT becomes smaller than that by VAR. Moreover, regardless of which initial classification is used, MSE decreases and converges as the number of iterations increases. The three MSEs converge to almost the same value, but the convergence speed is the fastest for DSIFT. The similar tendency has shown by the measurement of convergence characteristics based on three kinds of initial classification methods under various C and T_0. From these results, it can be concluded that the proposed algorithm can design the high-performance multi-class dictionaries regardless of initial classification method if the number of class update iterations is sufficiently large.

4.4.4 Coding performance

This section examines the rate distortion characteristics to evaluate the performance of our approach.

Table 4.2 details the performance (Bjontegaard metric[38]) of the proposed method at different C and T_0 values. In generating the data, the proposed method used dictionaries designed with 20 iterations, while the reference method to be compared used dictionaries designed under initial classification. As described later, there is an appropriate T_0 in the range of the target bit rate (compression ratio). In other words, a small T_0 is effective for an application used in a low bit rate environment, and a large T_0 is effective for an appli-

Table 4.2: BD-PSNR[dB] and BD-rate[%] of the proposed method against the conventional method without class update under the same C and T_0 as an anchor

T_0	C	***BasketballDrive***		***BQTerrace***		***Cactus***		***ChristmasTree***		***Kimono1***		***ParkScene***	
		PSNR	RATE	PSNR	RATE	PSNR	RATE	PSNR	RATE	PSNR	RATE	PSNR	RATE
3	32	0.85	-26.27	0.75	-26.32	0.78	-22.64	0.83	-40.58	0.67	-8.46	0.75	-17.27
	64	1.07	-32.27	0.95	-32.81	1.01	-29.59	0.95	-45.09	1.01	-13.15	0.9	-20.33
	128	1.03	-30.56	0.92	-30.75	0.99	-28.49	1.01	-48.26	0.98	-11.96	0.95	-21.23
5	32	0.79	-17.49	0.69	-17.17	0.75	-15.96	0.91	-26.59	0.43	-5.81	0.77	-12.97
	64	1.2	-25.88	1.02	-24.57	1.13	-23.61	1.16	-32.75	0.97	-14.06	1.05	-17.43
	128	1.12	-23.83	0.99	-23.6	1.09	-22.26	1.23	-34.7	0.92	-12.72	1.05	-17.32
7	32	0.84	-16.78	0.68	-15.03	0.77	-14.47	0.99	-22.2	0.46	-7.17	0.79	-12.12
	64	1.23	-23.56	1.04	-21.8	1.21	-21.97	1.35	-28.93	0.81	-12.07	1.18	-17.36
	128	1.16	-22.07	0.99	-20.35	1.13	-20.29	1.34	-28.3	0.89	-13.08	1.1	-16.2
9	32	0.85	-16.27	0.7	-14.62	0.78	-14.24	1.02	-19.56	0.56	-8.83	0.82	-12.2
	64	1.13	-20.59	0.99	-19.65	1.1	-19.09	1.4	-25.73	0.46	-6.7	1.18	-16.79
	128	1.13	-20.16	1.02	-19.69	1.13	-19.04	1.48	-26.32	0.77	-11.47	1.18	-16.45

cation used in a high bit rate environment. Also, there is an appropriate C depending on the implementation environment of the encoder / decoder. In other words, if the memory available to the encoder and decoder is large, we can use dictionaries designed with a large C, but if the memory is small, we can use only dictionaries designed with a small C. Thus, we can say there are many cases where dictionaries designed with fixed T_0 and C are used, to satisfy the requirements of the encoder / decoder implementation and application environment. Therefore, in Table 4.2, C and T_0 of the conventional method used as anchors are the same as C and T_0 of the proposed method, respectively. As shown in Table 4.2, regardless of the number of classes and sparsity parameters, our approach attained significantly better BD-PSNR and BD-rate performance than the conventional approach. From Table 4.2, it can be seen that BD-PSNR improved from 0.4 dB to 1.5 dB for various images. In addition, the performance improvement in BD-rate was 6% to 48%, and reducing sparsity parameter T_0 increased the bitrate reduction.

In addition, the performance of the proposed method is compared with that of the conventional method in which the encoding is performed with the number of classes that gives the best performance. First, the dictionaries in the conventional method are designed by setting the number of classes to 1, 4, 16, 32, 64, 128, and 256. After that, the rate-distortion characteristics when encoding with each dictionary are obtained. Next, using the encoding performance when $C = 1$ (i.e. single class) as an anchor, the number of classes, C_{best}, with the best BD-rate is determined for each T_0. The result is shown in Figure 4.10. Figure 4.10 shows the average BD-rate for the six test sequences, and almost the same results were also obtained for individual images. From Figure 4.10, we can see that C_{best} is 32. Next, summa-

rizes the BD-PSNR and BD-rate of the proposed method in Table 4.3, in which the anchor is the RD characteristic of the conventional method with C_{best}. Note that since $C_{\text{best}} = 32$, the row of $C = 32$ in Table 4.3 has the same value as Table 4.2. From the above considerations, it was confirmed that the coding performance of the proposed method also exceeded that of the conventional method using the number of classes which gives the best performance.

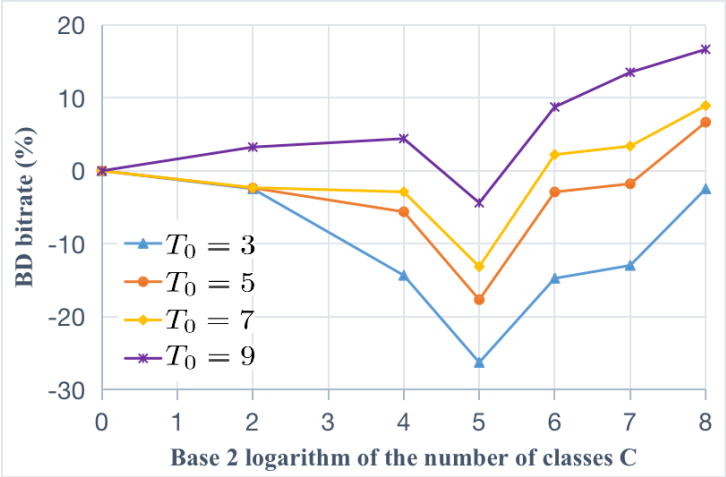


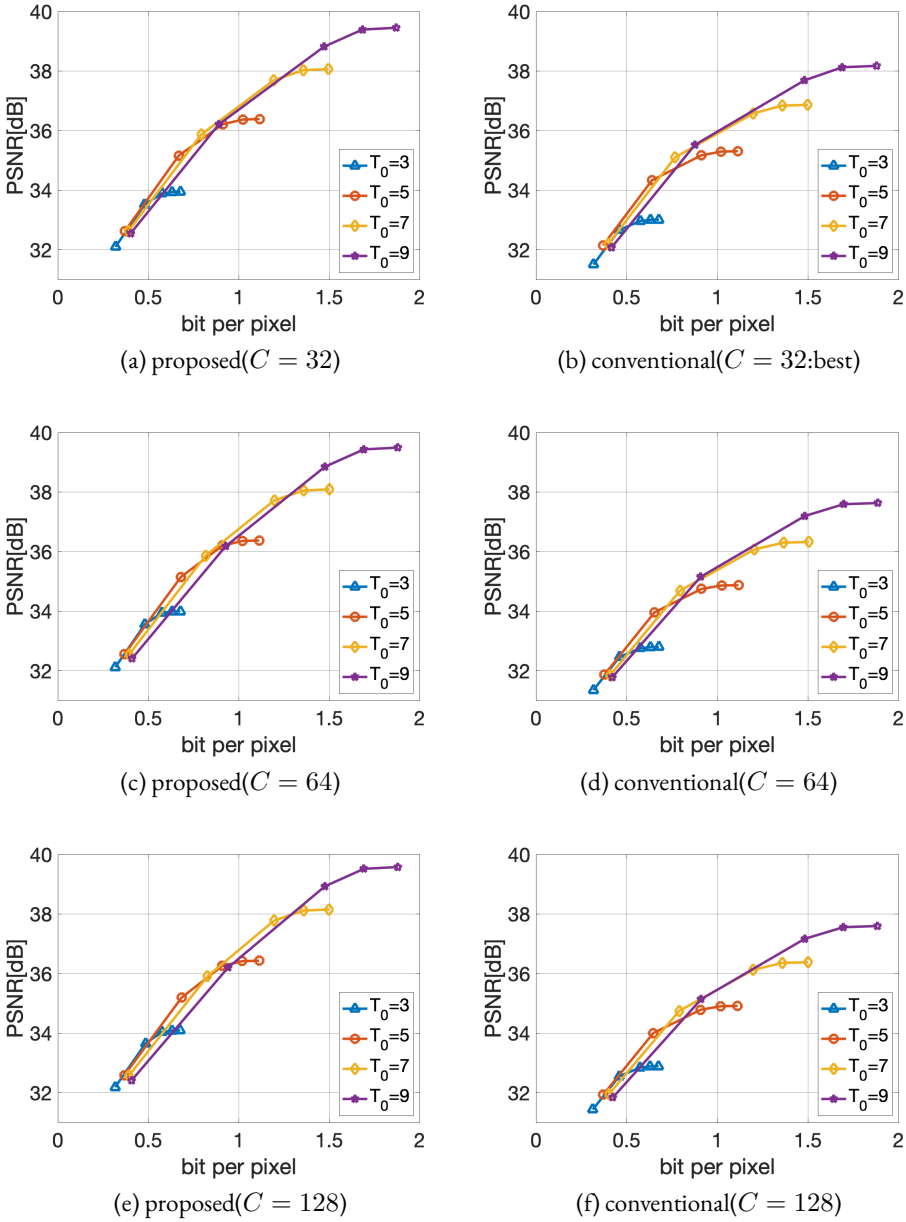
Figure 4.10: BD bitrate against $C = 1$ as an anchor (without class update)

Table 4.3: BD-PSNR[dB] and BD-rate[%] of the proposed method against the conventional method without class update under the best $C(= 32)$

Note that the values in the row for $C = 32$ are the same as in Table 4.2

T_0	C	*BasketballDrive*		*BQTerrace*		*Cactus*		*ChristmasTree*		*Kimono1*		*ParkScene*	
		PSNR	RATE	PSNR	RATE	PSNR	RATE	PSNR	RATE	PSNR	RATE	PSNR	RATE
	32	0.85	-26.27	0.75	-26.32	0.78	-22.64	0.83	-40.58	0.67	-8.46	0.75	-17.27
3	64	0.92	-27.50	0.80	-27.22	0.82	-23.22	0.88	-41.86	0.74	-9.42	0.80	-17.88
	128	0.99	-29.41	0.84	-27.68	0.90	-25.24	0.96	-45.50	0.84	-10.31	0.87	-19.29
	32	0.79	-17.49	0.69	-17.17	0.75	-15.96	0.91	-26.59	0.43	-5.81	0.77	-12.97
5	64	0.78	-16.88	0.66	-15.56	0.71	-14.08	0.88	-25.06	0.42	-5.44	0.74	-12.08
	128	0.81	-17.08	0.68	-15.44	0.75	-14.54	0.94	-26.64	0.48	-6.27	0.76	-12.45
	32	0.84	-16.78	0.68	-15.03	0.77	-14.47	0.99	-22.20	0.46	-7.17	0.79	-12.12
7	64	0.73	-13.78	0.64	-12.69	0.68	-11.65	0.95	-20.24	0.35	-4.90	0.71	-10.36
	128	0.78	-14.41	0.65	-12.34	0.70	-11.67	0.98	-20.29	0.41	-5.67	0.72	-10.40
	32	0.85	-16.27	0.70	-14.62	0.78	-14.24	1.02	-19.56	0.56	-8.83	0.82	-12.20
9	64	0.66	-11.72	0.60	-11.20	0.63	-10.17	0.92	-16.27	0.40	-5.86	0.68	-9.52
	128	0.68	-11.78	0.62	-11.05	0.63	-9.81	0.94	-16.4	0.41	-5.80	0.67	-9.21

Figure 4.11 and Figure 4.12 show the PSNR and bit rate of the test image "Cactus" and "ParkScene" measured under the condition of C= 32, 64, and 128. Figure 4.11 (a), (c) and (e) show the rate distortion characteristics of the proposed method, and Figure 4.11 (b), (d) and (f) show those of the conventional method. We can confirm the effectiveness of the proposed method relative to the conventional method under same C and T_0 by comparing Figure 4.11 (a), (c) and (e), with Figure 4.11 (b), (d) and (f), respectively. Also, note that Figure 4.11 (b) is the result of encoding with the number of classes that gives the maximum performance in the conventional method. We can see that the performances of the proposed method shown in Figure 4.11 (a), (c) and (e), are superior to the best performance of the conventional method shown in Figure 4.11 (b). A similar discussion is possible from the results in Figure 4.12. Figure 4.11 and Figure 4.12 show that as sparsity parameter T_0 fell, the PSNR saturated at a lower bit rate, even if a finer quantization level was used. Therefore, in order to obtain a high PSNR, encoding must use a larger number of non-zero coefficients and the dictionaries designed with large T_0 values. On the other hand, in the low bit rate environment, where the number of non-zero coefficients to be encoded increases, the quantization step width must be coarse, which leads to a decrease in PSNR. These results suggest that it is better to use the dictionaries designed with large T_0 values when high bit rates are possible and to use the dictionaries designed with small T_0 values if only low bit rates are available. We have confirmed that the same trend is observed for different test images and different C values. In order to realize this idea, a method of switching multiple dictionaries designed with various C and T_0 for each target compression ratio or each image/block will be suitable. Although the multiple dictionaries designed for various C and T_0 must be shared by the encoder and decoder, the method is considered to be useful as an advanced coding control method for both the conventional method and the proposed method. These advanced RD optimization method by adapting C/T_0 is an important subject in the future study.

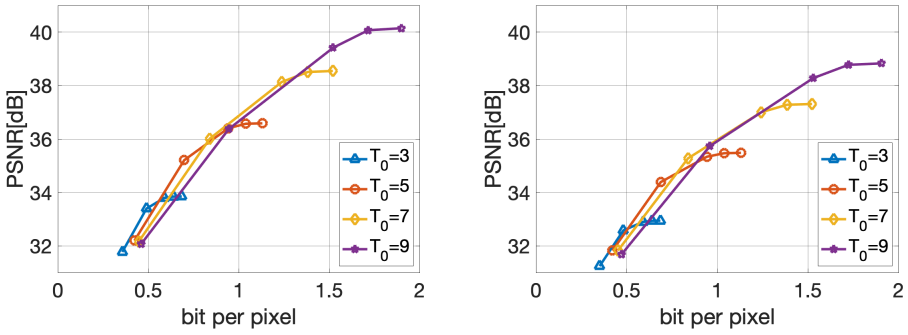


(a) proposed($C = 32$) (b) conventional($C = 32$:best)

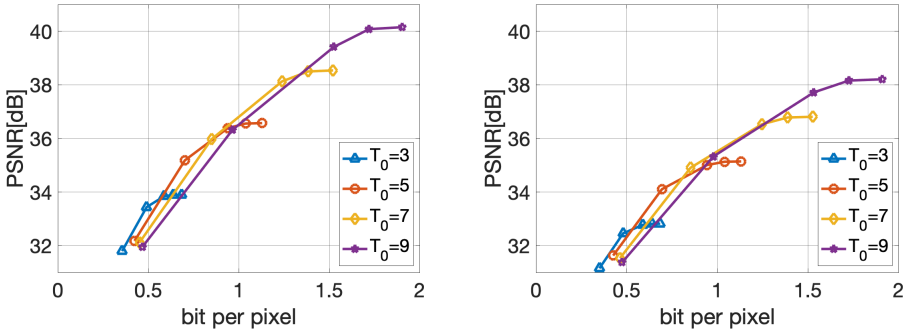
(c) proposed($C = 64$) (d) conventional($C = 64$)

(e) proposed($C = 128$) (f) conventional($C = 128$)

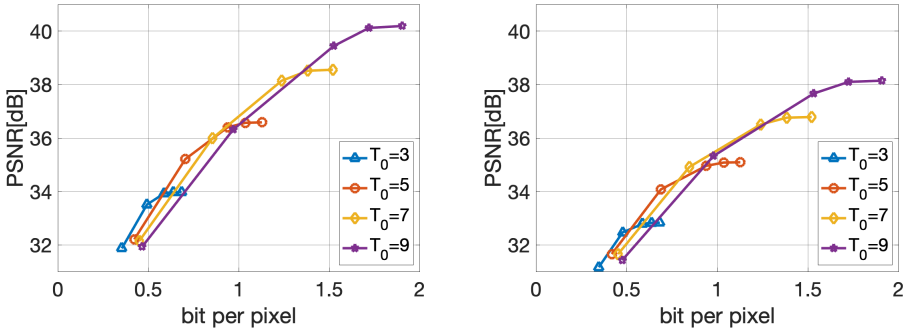
Figure 4.11: RD curves for Cactus. All results include overhead information for class indices



(a) proposed($C = 32$) (b) conventional($C = 32$:best)

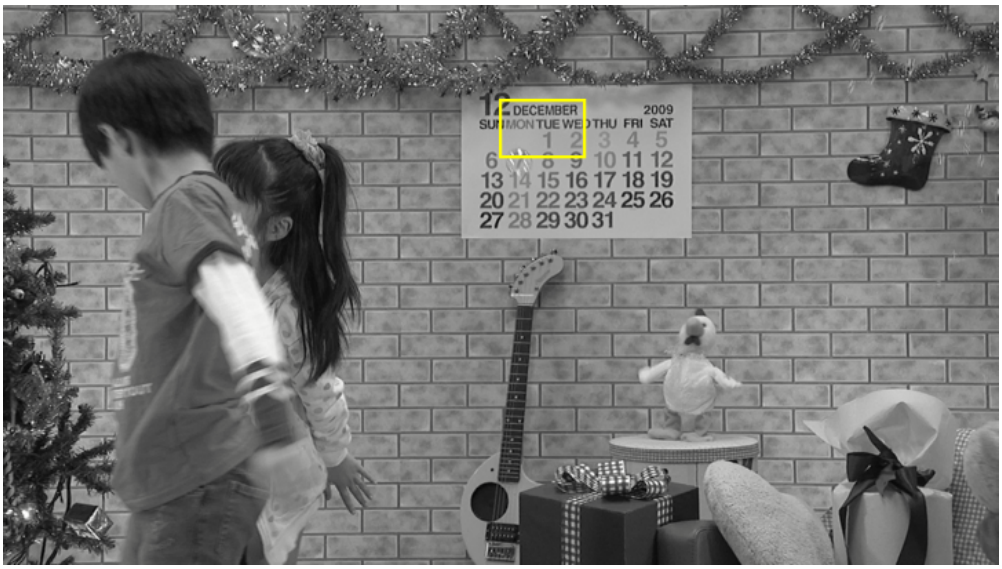


(c) proposed($C = 64$) (d) conventional($C = 64$)

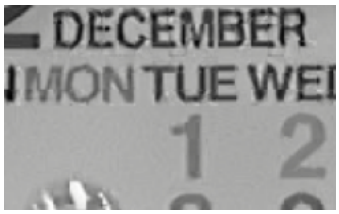


(e) proposed($C = 128$) (f) conventional($C = 128$)

Figure 4.12: RD curves for ParkScene. All results include overhead information for class indices

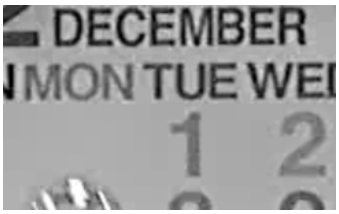


(a) ChristmasTree



(b) Original

(c) Without class-update



(d) With class-update

Figure 4.13: Perceptual quality comparison for ChristmasTree (0.53 bit/pel, $C = 128$, $T_0 = 9$)

Subjective image quality is also improved by the proposed method. Figure 4.13 compares the reconstructed images encoded at the same bit rate using dictionaries designed with $T_0 = 9$. It can be seen that the dictionary designed using class update can reconstruct detailed image structure with less visual degradation.

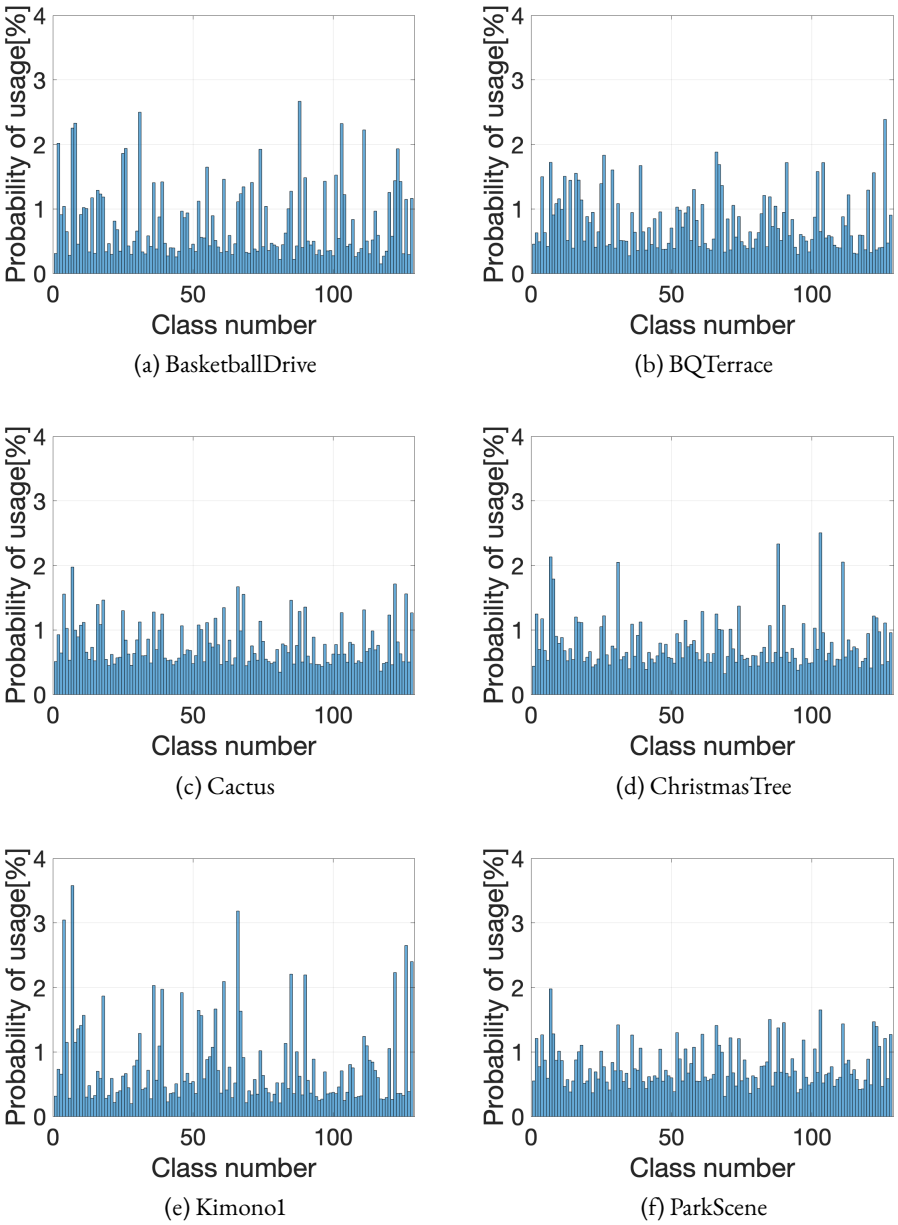
Figure 4.14 shows a histogram of class index selection for each image when an image is coded using a dictionary designed under the condition of $C = 128$ and $T_0 = 3$. From Figure 4.14, it can be seen that the selection ratio of classes differs with the image, and that image reconstruction does trigger switching to the appropriate dictionary according to the distribution of the local features of the image to be encoded. Figure 4.15 shows the relationship between the feature of the bases included in some dictionaries and the feature of the blocks that selected each of those dictionaries. Figure 4.15 (a) is a part of image "Park Scene". Figure 4.15 (b), Figure 4.15 (d) and Figure 4.15 (f) show the blocks using the dic-

tionaries shown in Figure 4.15 (c), Figure 4.15 (e) and Figure 4.15 (g), respectively. We can see that the blocks in Figure 4.15 (b), Figure 4.15 (d) and Figure 4.15 (f) contain vertical, diagonal and horizontal edge, respectively, and the selected dictionary contains many bases that reflect the block feature. These confirm the effectiveness of the multi-class dictionary approach. Also, since there is no class index that is rarely used, it can be said that the number of classes is not excessive.

Next, we consider the local correlation of class indices. If the class indices of neighboring blocks are highly correlated, applying the following rules in code assignment may reduce the total code length. Let C_P be the class index of the target block, C_A be the class index of its left neighboring block, and C_B be the class index of its upper neighboring block. The code assignment rule is:

- "00" if $C_P = C_A$
- "01" if $C_P = C_B$
- "1"+"fixed length code with $\lceil \log_2 C \rceil$ bit" otherwise.

The number of bits yielded by the above rule is smaller than that yielded by fixed length coding only when the probabilities of $C_P = C_A$ or $C_P = C_B$ are larger than 0.25 ($C = 32$), 0.2 ($C = 64$), 0.167 ($C = 128$), respectively. I measured the probability that the class index of a block to be coded is the same as the class index of its left or upper neighboring blocks for the six test images in Figure 4.5 under $T_0 = 3$. They were 0.118, 0.076, 0.054 for the case of $C = 32$, $C = 64$ and $C = 128$, respectively. This result suggests that the class index has only slight local correlation, and that using a variable length code to the class has little benefit. Therefore, as described in Section 4.3, it is appropriate to assign a fixed length code to each class index.

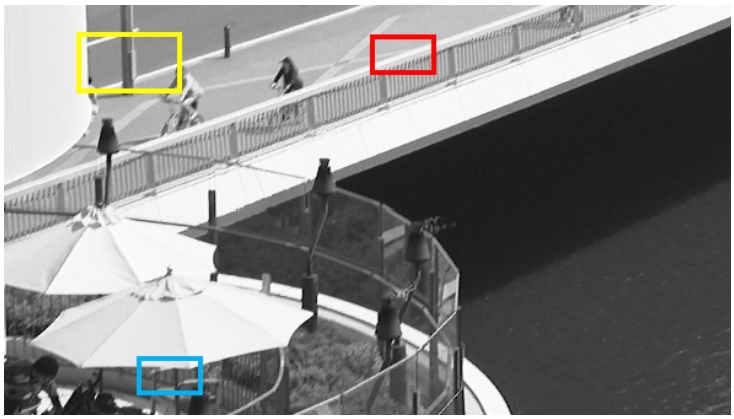


(a) BasketballDrive (b) BQTerrace

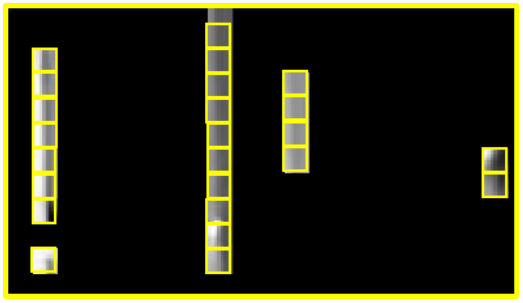
(c) Cactus (d) ChristmasTree

(e) Kimono1 (f) ParkScene

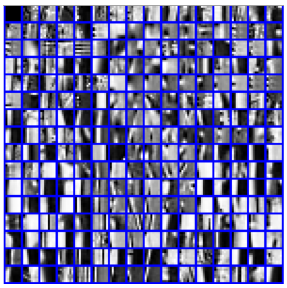
Figure 4.14: Class selection probability ($T_0 = 3, C = 128$)



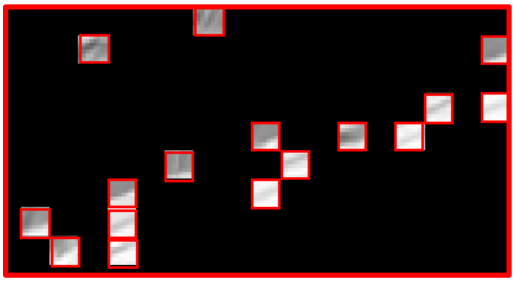
(a) A part of "BQTerrace"



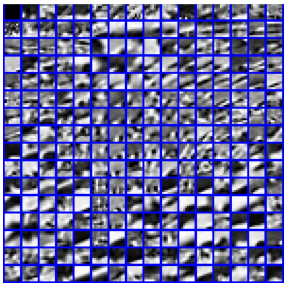
(b) Block corresponding to dictionary (c)



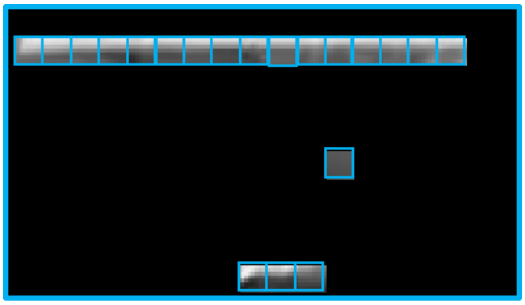
(c) Dictionary ♯25



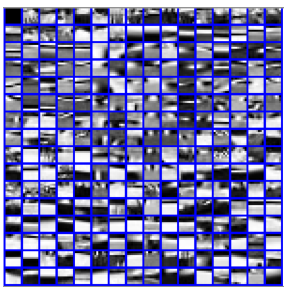
(d) Block corresponding to dictionary (e)



(e) Dictionary ♯29



(f) Block corresponding to dictionary (g)



(g) Dictionary ♯88

Figure 4.15: Example of relationship between the feature of blocks and the selected dictionary

4.5 Summary

In this chapter, we first review the research related to the multi-class sparse dictionary applied to image coding, and introduced a framework for multi-class dictionary design. Since such dictionaries did not consider the relevance of classification effects and dictionary design, therefore in section 4.2, I proposed a multi-class K-SVD method that considers interdependency of classification performance and dictionary design. In the proposed method, after multiple dictionaries are designed by K-SVD, sparse coding for each training vector is per-formed using all of the dictionaries. As a result, the training vector is reclassified into the class that best approximates it. By iteratively performing the dictionary design stage and the class update stage, it is possible to design dictionaries that enable more efficient sparse representation. The way to hire this method for image coding is introduced in section 4.3, and the coding experiments are detailed in section 4.4. Experiments on still images revealed that the proposed algorithm gives a significant coding gain compared to the conventional method based on fixed classification with predetermined features.

Entropy Coding Method for Sparse Coefficients

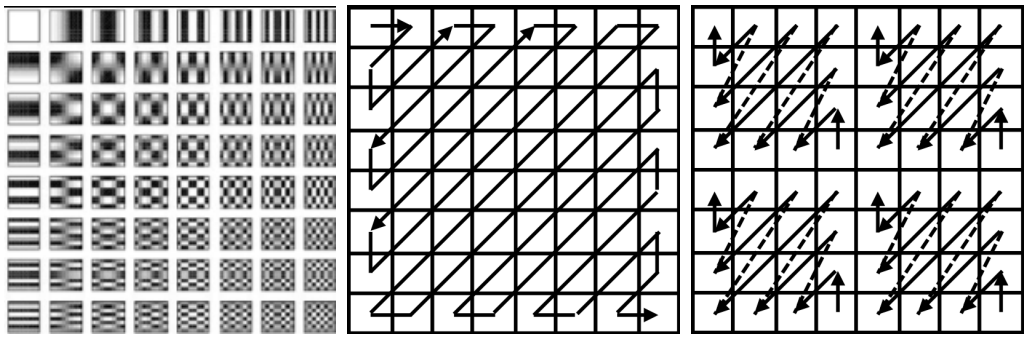
5

Contents

When the sparse coding is applied to image compression, the problem is how to encode the nonzero coefficients distributed in sparse. The statistical properties of sparse nonzero coefficients have been analyzed in some previous studies. In [45], it has been experimentally reported that the atom indices to indicate the occurrence position of nonzero coefficient can be approximated by uniform distribution, and nonzero coefficient levels can be approximated by Laplacian distribution. However, it is not clear how the atom indices and the nonzero coefficient levels in a block are related to the number of nonzero coefficients in the block. Also, a detailed analysis of the relationship between a nonzero coefficient level and its corresponding atom's feature has not been performed. For more efficient entropy coding design, it is necessary to analyze statistical properties of nonzero coefficients in more detail.

5.1 Methods of Entropy Coding

In image coding, it is necessary to make symbols to be coded into binary codes. This procedure is called entropy coding, and various kinds of variable length coding (VLC) based on the occurrence probability of symbols are utilized. By assigning fewer bits to encode more frequently occurring symbols, the total amount of bits used to encode the all symbols can be reduced.



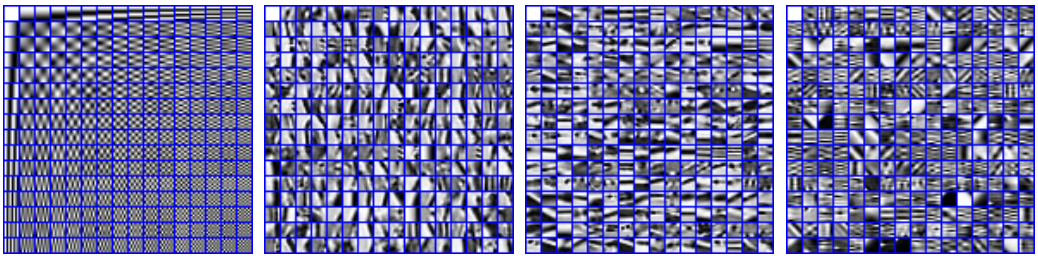
(a) DCT atom (b) Scan order in JPEG (c) Scan order in H.265/HEVC

Figure 5.1: DCT atom and scan order

5.1.1 Code assignment techniques for transform coefficients

Here, we review some code assignment techniques for transform coefficients. Discrete Cosine Transform (DCT) is one of the most popular techniques used today in video compression schemes. Figure 5.1(a) shows the 8×8 array of atom images for the two dimensional DCT. DCT converts a set of pixels in a block into the weighted sum of DCT atoms. The weighting factors are called DCT coefficients. Statistically, the magnitude of the DCT coefficients for low frequency atoms are greater than that for high frequency atoms. Also, by quantizing the coefficients, many DCT coefficients for high frequency atoms will be zero.

This property is used to perform efficient entropy coding by appropriately setting the scan order of DCT coefficients. The order of zigzag scan in JPEG and MPEG-2 is shown in Figure 5.1(b). The first coefficient of each block obtained as a result of zigzag scan is called the DC coefficient while the other coefficients are called AC coefficients. For AC coefficients, a variable length code is assigned for the pair of a nonzero coefficient and its preceding zero-run length[56, 57]. An End-of-Block (EOB) at the end of each block indicates the rest of the coefficients of the block are all zero, and it enables to represent long consecutive zeros effectively. In H.265/HEVC, quantized DCT coefficients are coded as follows. They are scanned diagonally to form a 1D array as shown in Figure 5.1(c). The context adaptive binary arithmetic coder (CABAC) encodes the last position of nonzero coefficients, a significance map indicating the positions of nonzero coefficients, and the quantized coefficient level values[58, 59].



(a) (b) (c) (d)

Figure 5.2: Examples of multiclass dictionaries designed by K-SVD
(a) overcomplete DCT, (b, c, d) Dictionaries designed by K-SVD

5.1.2 Related works

In the case of complete DCT, the frequency characteristics of each atom are known, and they are regularly arranged. The relative relationship between the characteristics of each atom and the magnitude of the transform coefficient corresponding to each atom is also clarified. Therefore, by setting the scan order as shown in Figure 5.1 based on these characteristics in advance, the number of occurred bits can be reduced effectively. Also, for the atoms based on complete DCT, the international standard methods H.264/AVC and H.265/HEVC have adopted a method of switching the code table for each block using the number of non-zero coefficients as a context[58, 60]. On the other hand, each atom of the overcomplete dictionary designed by K-SVD does not have regular frequency characteristics like DCT. Thus, it has not been clarified what kind of atom has a large non-zero coefficient. Also, it has not been clarified how the coefficient quantization level distribution changes with the number of non-zero coefficients in the block. Therefore, in order to perform entropy coding for sparse coefficients as efficient as the conventional method, we need to clarify the statistical properties of the sparse coefficients and to clarify how to reorder the sparse coefficients based on the findings.

Figure 5.2 shows some examples of dictionaries designed by multi-class K-SVD. It is important to note that the atoms in a dictionary designed by K-SVD are not necessarily arranged in frequency order like DCT, and the atoms with different properties appear randomly.

Several entropy coding for sparse representations have also been studied. In image coding using sparse representation, OMP is performed using a dictionary for each block to be coded, and at most T_0 nonzero coefficients are calculated. All other coefficients are zero. For the entropy coding of the sparse representation, the indices of atoms corresponding nonzero

coefficients after quantization and the nonzero coefficients levels are encoded. In the conventional studies, it has been clarified that the histogram of the atom's indices is approximated to a uniform distribution, the histogram curve for the quantized coefficient levels is approximated to the Laplacian distribution[45, 61]. In [61], it is shown that the nonzero coefficient in the case of sparse representation by overcomplete DCT becomes Laplacian distribution. Based on these features, a fixed length code was assigned for the atom index coding, and Huffman code or a truncated unary code combined with an Exponential-Golomb code was employed to encode the quantized coefficient level[45, 61]. On the other hand, instead of assigning a code directly to an index, a method of assigning a Huffman code to a zero-run length (i.e. the number of consecutive zero coefficients between nonzero coefficients) has also been studied[1].

However, in the conventional researches, the relationship between the atom indices corresponding the nonzero coefficients in a block and the number of nonzero coefficients of the block has not been clarified. Also, the relationship between the probability distribution of the nonzero coefficient levels and the number of nonzero coefficients of the block has not been clarified. In addition, the detailed analysis of the relationship between the magnitude of nonzero coefficient level and the feature of the corresponding atoms has not been conducted. Therefore, there is room for improving the conventional code assignment procedure by using the number of nonzero coefficients and feature of the atoms as a context. In the next section, the statistical properties of nonzero coefficients are analyzed in detail from theoretical and experimental viewpoints for sparse representation of images, and I propose an efficient entropy coding scheme for sparse coefficients.

5.2 Statistical Properties of Sparse Coefficients

In this section, the statistical properties of the sparse coefficients are analyzed in detail for the entropy coding scheme design. The analysis in Section 5.2 is carried out theoretically and experimentally. A set of small blocks extracted from six types of images, "BQTerrace", "BasketballDrive", and "Cactus", "ChristmasTree", "Kimono1" and "ParkScene" from the MPEG test sequence are used for statistical analysis, where these images are also used as test data for the experiments in Section 4.4.1.

The sparse coefficients to be encoded can be illustrated as in Figure 5.3. First, the image is divided into small blocks of $\frac{\sqrt{L}}{2} \times \frac{\sqrt{L}}{2}$. Next, for each small block, OMP is performed on the dictionary designed by K-SVD to obtain T_0 sparse coefficients. The dimension of a dictionary is L. After quantization, I obtain sparse coefficients to be encoded for each small

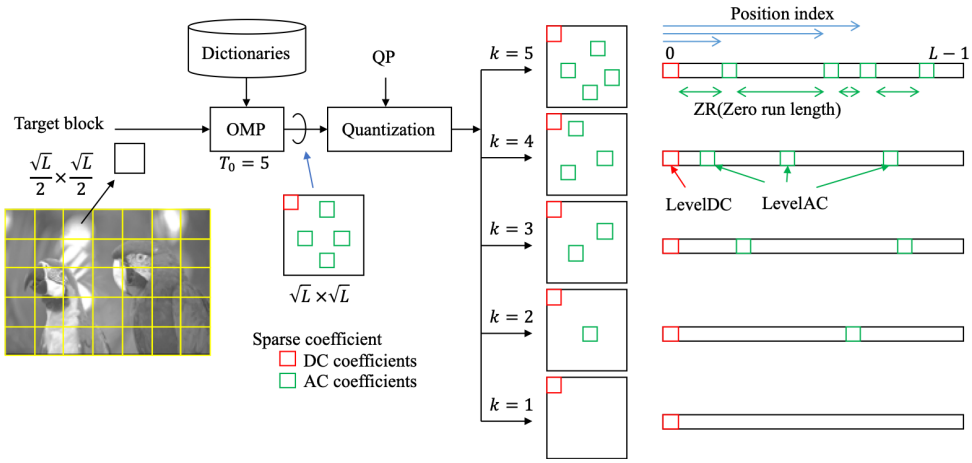


Figure 5.3: Sparse coefficients to be coded

block. Let k be the number of nonzero coefficients after quantization. Here, all DC coefficients are quantized by step one and they are always encoded. Also, nonzero AC coefficients are quantized by the quantization step QP. The number of nonzero AC coefficients after quantization is at most $T_0 - 1$. Set the number of blocks in which the number of nonzero coefficients to be coded becomes k among all blocks of the image as $N(k)$. $N(1)$ means the number of the blocks represented by DC coefficients only. The total number of blocks in the whole image, N, is $N = \sum_{k=1}^{T_0} N(k)$, and the number of DC coefficients in the image, N_{DC}, is equal to N. In addition, the number of nonzero AC coefficients in the whole image, $N_{\mathrm{nonzeroAC}}$, is expressed by:

$$N_{\mathrm{nonzeroAC}} = \sum_{k=2}^{T_0} (k-1)N(k) \tag{5.1}$$

5.2.1 Syntax of sparse coefficients coding

Figure 5.4 shows the sparse coefficient coding syntax analyzed in this study. The information required for each block to be encoded are, $class\ No.$: a class number indicating which of dictionaries is used, k: the number of nonzero coefficients in the block, $coef_{DC}$: a weighting factor for DC atom, and $coef_{AC}$: weighting factors for AC atoms. Also, the number of nonzero AC coefficients is $k - 1$, and it is necessary to encode atom indices and quantized coefficient levels for each nonzero AC coefficient. In this study, in order to perform code allocation adaptively by the number of nonzero coefficients for each block, the

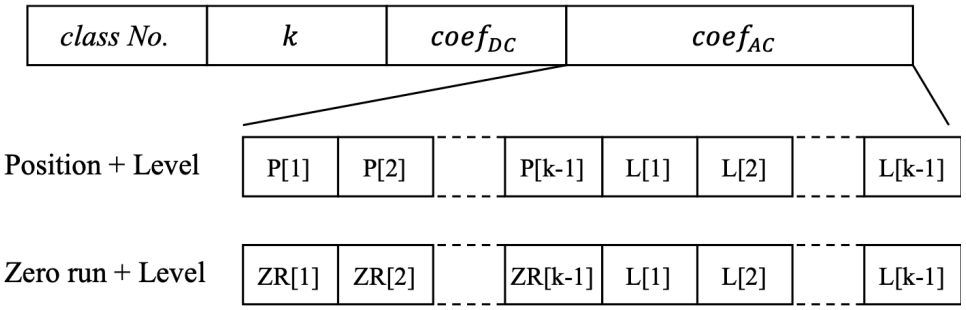


Figure 5.4: Bit stream structure for sparse coefficients

number of nonzero coefficients in a block, k, is encoded prior to the atom indices of the nonzero coefficients and the quantized coefficients level. For the syntax of AC coefficients, the atom indices for nonzero coefficients and the nonzero quantized coefficients level are encoded. Regarding the atom indices, we will consider two kinds of methods: direct encoding of indices and a method of using zero run length between indices of nonzero coefficients.

5.2.2 Nonzero coefficients distribution and entropy

In the conventional research[45, 61], the atom indices for nonzero coefficients after quantization and the nonzero quantized coefficient levels are coded independently, and any adaptation of code assignment depending on the number of sparse coefficients in the block and the feature of the atoms has not been studied. In this subsection, I first unify the symbols of all blocks based on the conventional method and analyze the statistical properties of the nonzero coefficients and the amount of generated bits. In this study, the amount of generated information is analyzed based on the entropy calculated from the occurrence probability of the symbols to be coded. The total amount of generated bits for the whole image is expressed as:

$$Bit_{all} = Bit_{class} + Bit_k + Bit_{DC} + Bit_{AC} \tag{5.2}$$

Here, Bit_{class}, Bit_k, Bit_{DC} and Bit_{AC} are the amount of generated bits for class number, the number of nonzero coefficients, DC coefficient, AC coefficient, respectively. The amount of each code bits can be calculated as follows.

First, a class number can be expressed as a fixed-length code of $\log_2 C$ bits per block, where C is the number of classes. The amount of generated bits in the whole image, Bit_{class}, can be calculated as $Bit_{class} = N \times \log_2 C$.

Next, to calculate the number of bits for the number of nonzero coefficients, it is neces-

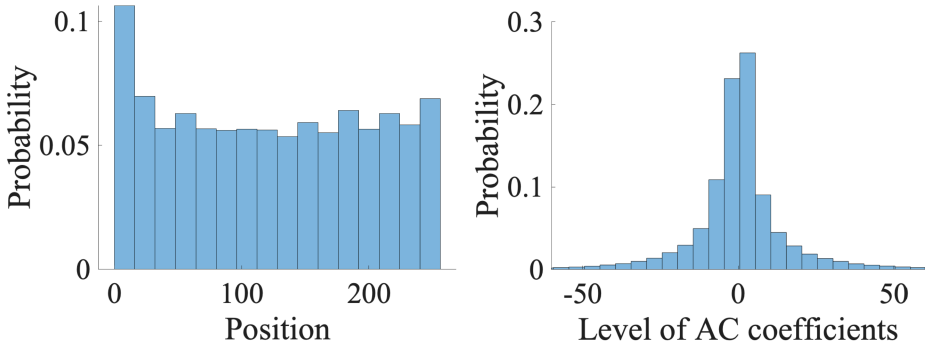


Figure 5.5: Probability histograms of (a) position index and (b) magnitude of nonzero quantized AC coefficients

sary to consider the distribution of the occurrence probability $p(k)$. $p(k)$ changes with the quantization step QP for the coefficients. When the QP becomes smaller, the occurrence probability of large k increases, and as the QP becomes coarser, the occurrence probability of small k increases. The amount of bits for the number of nonzero coefficients in the whole image is calculated as $Bit_k = E_k \times N$, where E_k is the entropy of $p(k)$ as shown following equation.

$$E_k = -\sum_{k=1}^{T_0} p(k) \log_2 p(k) \tag{5.3}$$

The amount of bits generated for the DC coefficient is calculated as follows. Since the DC coefficients reflect the average value of the block, there is a high correlation between the DC coefficients of adjacent blocks. Therefore, DPCM is performed based on the difference with the previous block. Since the probability distribution of the difference signal is approximated as a Laplacian distribution centered at zero, the total amount of bits is calculated as $Bit_{DC} = E_{DC} \times N$, where E_{DC} is an entropy based on the occurrence probability of differential DC values.

The amount of bits generated for nonzero AC coefficient is calculated from the distribution of their atom indices and coefficient levels. Figure 5.5(a) shows a histogram of atom indices for nonzero AC coefficients measured when sparse coding is performed on the test images by setting $T_0 = 7$ and QP$= 16$. From the results, the occurrence probability of atom indices for nonzero coefficient is almost uniform. Similar measurements were performed for various combinations of T_0 and QP, and a chi-square test was performed for each case. As a result, we could confirm the uniformity of the probability distribution of atom

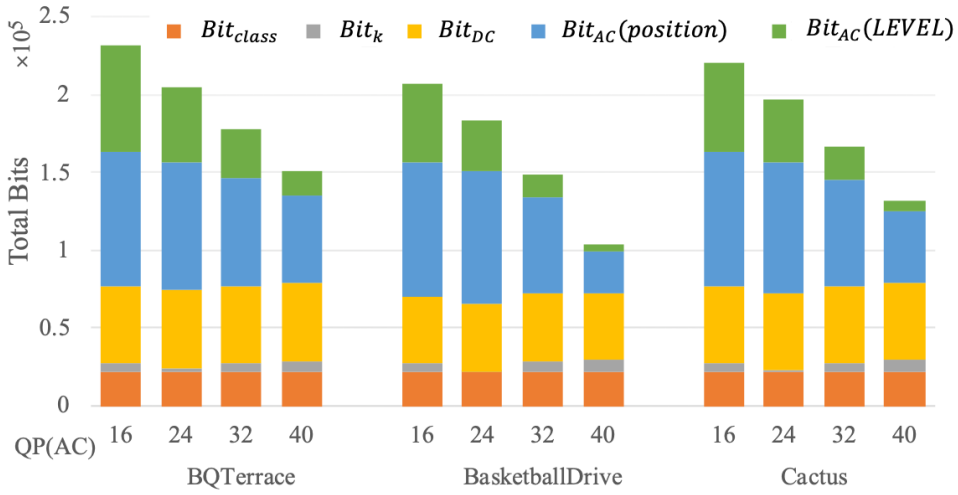


Figure 5.6: Number of bits generated

indices as in the conventional study[45]. When uniformity of the occurrence probability distribution of atom indices can be assumed, $\log_2 L$ bits are needed per one atom index theoretically. Therefore, the total amount of bits for atom indices in the whole image, Bit_I, is $Bit_I = \log_2 L \times N_{\text{nonzeroAC}}$. Also, Figure 5.5(b) shows the distribution of nonzero quantized AC coefficient levels, which can be approximated by the Laplacian distribution centered on zero. Note that there is no zero coefficient. Coarse quantization concentrates the occurrence probability distribution to smaller levels and increases the number of zero coefficients. The amount of bits for nonzero coefficient levels in the whole image, Bit_L, is $E_L \times N_{\text{nonzeroAC}}$, where E_L is the entropy of the nonzero AC coefficient levels. The total amount of bits for nonzero AC coefficient in the whole image, Bit_{AC}, is calculated as the sum of Bit_I and Bit_L.

Figure 5.6 shows the amount of bits generated in the whole image measured by changing QP. The coefficient level becomes smaller when the coarse quantization step is used, so the amount of bits for AC coefficient levels decreases. Similarly, when coarse quantization step is used, the number of nonzero quantized AC coefficients decreases, so the amount of bits for atom indices decreases. Since the quantization step for DC coefficients is always one, Bit_{DC} is constant regardless of the quantization parameter QP for AC coefficients. Bit_k shows a slight increase or decrease because the distribution of the number of nonzero AC coefficients changes depending on the magnitude of QP. Bit_{class} is constant because it is determined only by the number of classes. From Figure 5.6, it is clear that reducing the amount of bits for expressing the AC coefficient is very significant.

In order to reduce the amount of generated bits for the nonzero AC coefficients, it

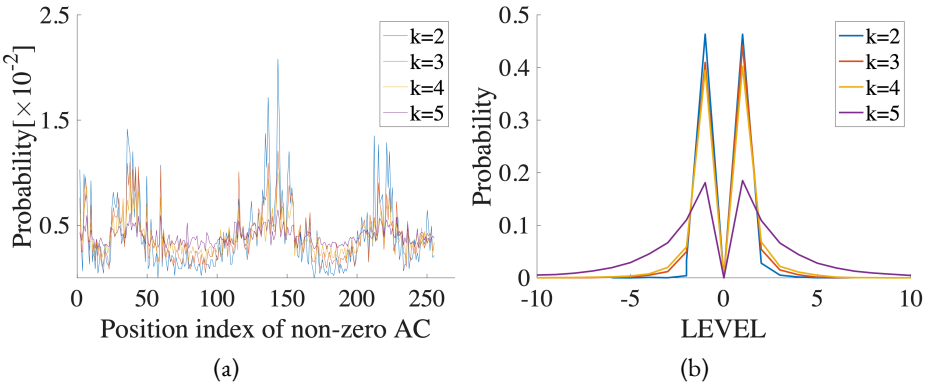


Figure 5.7: Probability histograms of (a) position index and (b) magnitude of nonzero quantized AC coefficients, after categorizing based on k

is possible to divide the nonzero AC coefficients into multiple categories according to the number of nonzero coefficients in the block and perform code allocation suitable for each category. Theoretically, if the symbols can be separated into multiple categories so that their occurrence probability distributions are as different each other as possible, the total code amount can be reduced.

Figure 5.7 shows the distribution of atom indices for nonzero AC coefficients and the distribution of AC coefficient levels, after categorizing based on the number of nonzero coefficients in the block. As shown in Figure 5.7, it is clear that the information symbols separation by k has little effect because the probability distribution of the atom's indices corresponding nonzero coefficients is almost same regardless of the value of k. On the other hand, since the occurrence probability of nonzero quantization level numbers show different distributions depending on k, it is considered to be significant to perform the symbol separation by k.

5.3 Proposal of Sparse Coefficients Entropy Coding

5.3.1 Sparsity adaptive sparse coefficient coding

Another way to represent atom indices of nonzero coefficients is to use the number of zero coefficients (i.e. zero run length) preceding nonzero coefficients[1]. The statistics of zero run length is analyzed when L coefficients are divided by k nonzero coefficients as shown in Figure 5.8. This problem can be solved theoretically as a consequence of the broken stick

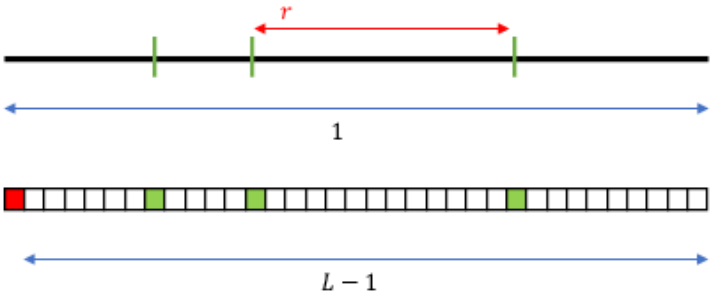


Figure 5.8: The probability density function of r, the length of any divided segments

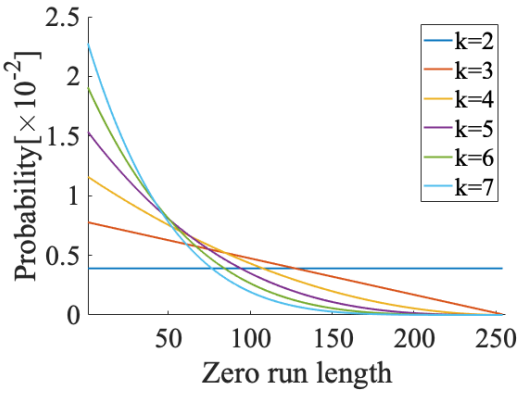


Figure 5.9: The theoretical probability distribution of zero run length

problem[62], which is an analysis problem concerning the probability distribution of the length for a piece of sub segments when the line segment of length 1 is divided by $n-1$ random points. The probability density function of the length $r(0 \leq r \leq 1)$ of any divided segments is:

$$g(r) = (n-1)(1-r)^{n-2} \tag{5.4}$$

The probability $P(r_0)$ that the length of each segment becomes $[r_0, r_0+\epsilon)$ is obtained by integration of equation(5.4) as:

$$\begin{aligned} P(r_0) &= \int_{r_0}^{r_0+\epsilon} g(r)dr \\ &= \left[-(1-r)^{n-1}\right]_{r_0}^{r_0+\epsilon} \\ &= (1-r_0)^{n-1} - \Big(1-(r_0+\epsilon)\Big)^{n-1} \end{aligned} \tag{5.5}$$

Applying the above analysis to the distribution of sparse coefficients, the length of the original line segment corresponds to the dimension L of a dictionary, and the length of each divided segment corresponds to the zero-run length. Figure 5.9 shows the theoretical probability distribution of zero run length when the length of the original line segment is set to $L = 256$. The occurrence probability is found to be a distribution based on an exponential function. In Reference [1], code design is performed by integrating the occurrence probabilities, that is, without classification by the number of nonzero coefficients. However, from Figure 5.9, since the parameters of exponential function clearly differ depending on the number of nonzero coefficients in the block, it can be expected that more efficient code assignment for zero run length becomes possible by categorizing nonzero coefficients by k. The entropy of the zero-run length is:

$$E_{run}(k) = \sum_{i=0}^{L-1} P(i/L) \log_2 P(i/L) \tag{5.6}$$

where $P(i/L) = (1 - i/L)^k - \Big(1 - (i/L + 1/L)\Big)^k$. The amount of bits to represent the atom indices in the whole image is calculated as:

$$Bit_I = \sum_{k=2}^{T_0} E_{run}(k-1)N(k)(k-1) \tag{5.7}$$

5.3.2 Adaptive coding by atom features

It is known that the features of the atoms appearing in the dictionary designed by K-SVD are strongly influenced by the features of the training samples, and they are different from general atoms such as DCT. Figure 5.10 shows the Fourier power spectrum of each atom for the four dictionaries shown in Figure 5.2. The center of each spectral image corresponds to the DC component, and the longer the distance from the center, the higher the frequency. For comparison, the power spectrum for complete DCT was added as shown in Figure 5.10(e). From Figure 5.10, in the dictionary consisting of atoms with regular frequency arrangement such as DCT and overcomplete DCT (Figure 5.10 (a), (e)), each atom complements each other so as to cover all frequency bands. On the other hand, the overcomplete dictionary designed by K-SVD (Figure 5.10 (b), (c), (d)) does not necessarily consist of atoms that cover all frequency bands. It can be seen that it is composed of atoms that can express a specific frequency band in more detail. When expressing images using a training-based dictionary, there have been no studies investigating the dependency between the characteristics of the weighting factors and the features of the atoms. If there is a correlation between some fea-

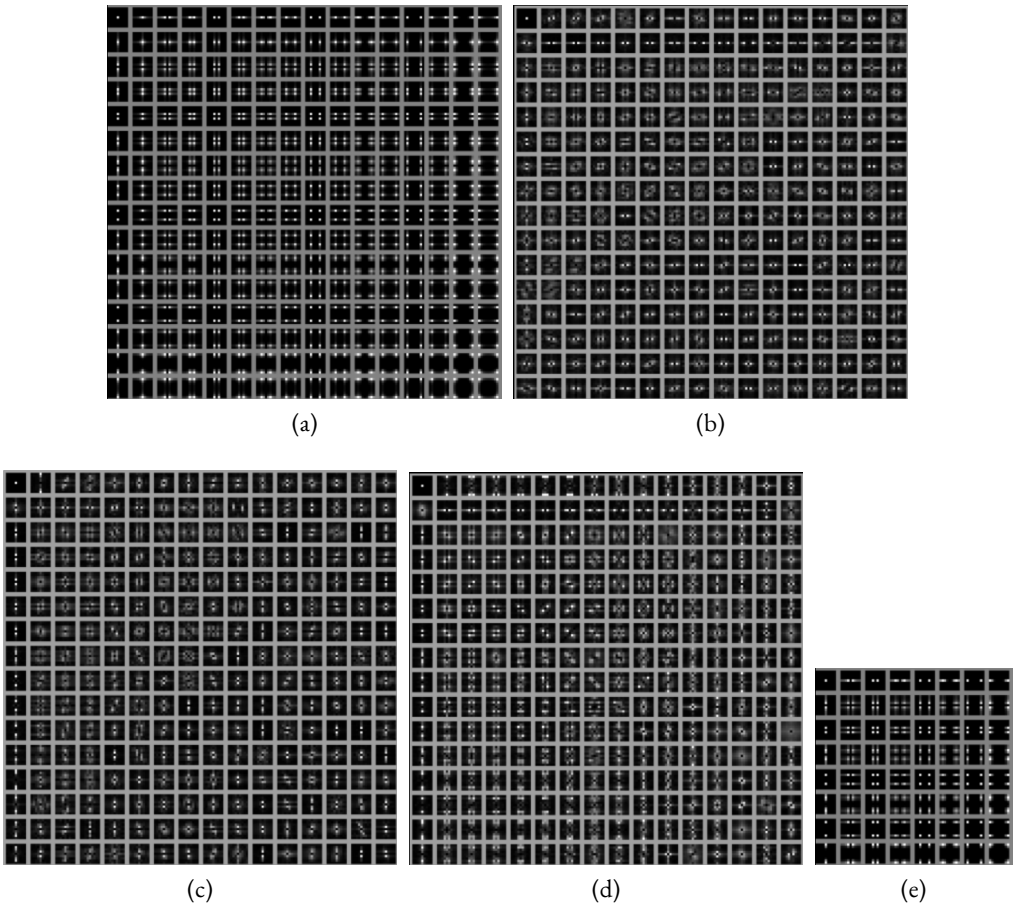


Figure 5.10: Power spectrum for atoms
(a) overcomplete DCT (Figure 5.2(a)), (b) K-SVD(Figure 5.2(b)), (c) K-SVD(Figure 5.2(c)), (d) K-SVD(Figure 5.2(d)), (e) complete DCT

tures of atom and weighting factors, it is possible to reduce the number of generated bits by adaptively changing the code assignment to the weighting factors with the atom's features as the context.

Therefore, we first investigated the relationship between some features of atoms and the magnitude of the weighting factors. Let $b(i, j)$ be an $M \times M$ atom in a dictionary designed by K-SVD. The following four features are investigated as the features of each atom,

- Fourier transform:

$$F_1(th) = \frac{\sum_{|u|+|v|\leq th} |B(u,v)|^2}{\sum_{u,v} |B(u,v)|^2}$$

where $B(u,v)$ is the Fourier power spectrum of $b(i,j)$, $-M/2 \leq u \leq M/2, -M/2 \leq v \leq M/2$.

- Discrete cosine transform:

$$F_2(th) = \frac{\sum_{(u+v)\leq th} |C(u,v)|^2}{\sum_{u,v} |C(u,v)|^2}$$

where $C(u,v)$ is the DCT coefficients of $b(i,j), 0 \leq u \leq M-1, 0 \leq v \leq M-1$.

- Total variation:

$$F_3 = \sum_i \sum_j \Big(|b(i+1,j) - b(i,j)| + |b(i,j+1) - b(i,j)| \Big)$$

- Number of strong edge:

$$F_4(th) = \sum_i \sum_j \Big(m_H(i,j) + m_V(i,j) \Big)$$

where

$$m_H(i,j) = \begin{cases} 1, & \text{if } |b(i+1,j) - b(i,j)| > th \\ 0, & \text{else} \end{cases}$$

$$m_V(i,j) = \begin{cases} 1, & \text{if } |b(i,j+1) - b(i,j)| > th \\ 0, & \text{else} \end{cases}$$

Figure 5.11 shows the correlation between each feature of atoms and the magnitude of the nonzero AC coefficient. Figure 5.11 also shows the average and the standard deviation of the absolute value of nonzero AC coefficients generated for the atoms in each section after dividing the feature quantity into 16 sections. In Figure 5.11, the results show the case where the parameter th for each feature value is set so that the correlation coefficient becomes the highest. There is a significant correlation between these four feature values and the magnitude of nonzero AC coefficients. Therefore, if we adapt the code assignment to the nonzero AC coefficient levels according to the feature of their corresponding atoms, the amount of generated bits can be reduced. Also, from the observation in Figure 5.11, we can consider that more efficient code assignment for the length of zero runs is performed by reordering the atoms so that the coefficients with large absolute values are scanned first. Figure 5.12 shows the examples of the atoms reordered by their features. Because the reordering of atoms concentrates nonzero AC coefficients at the start of the scan, so the probability of

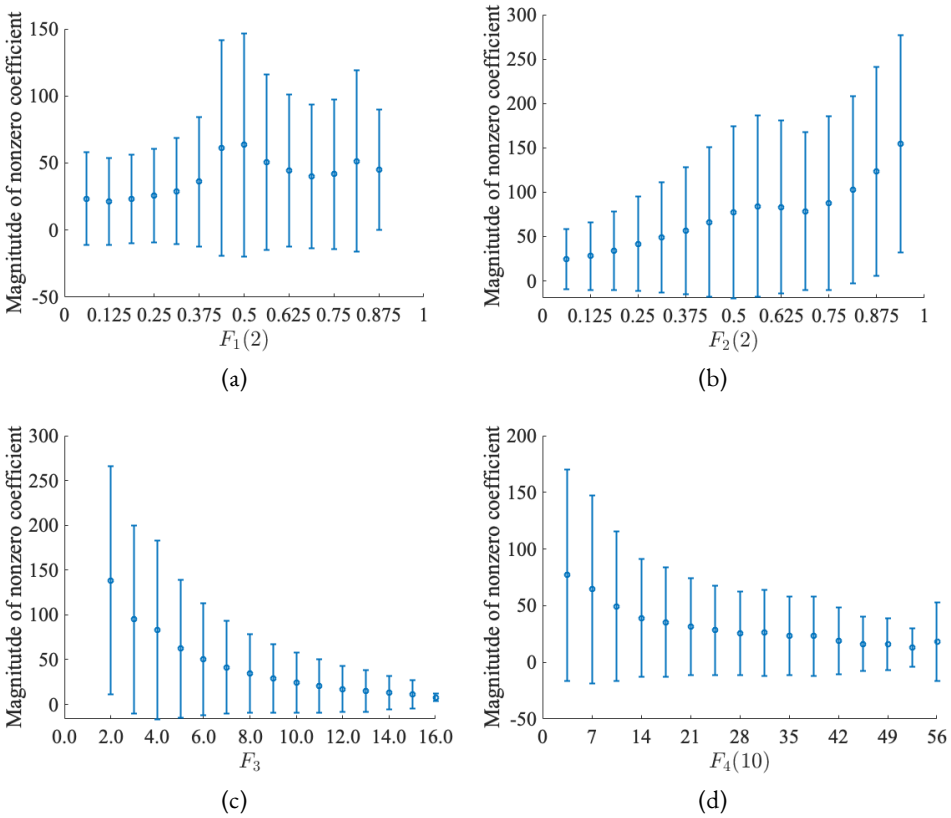
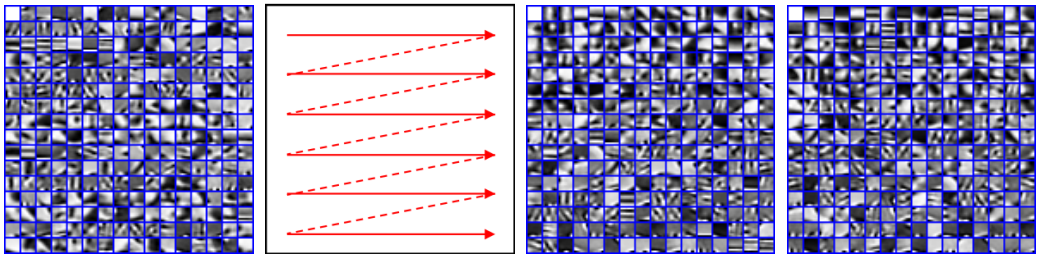


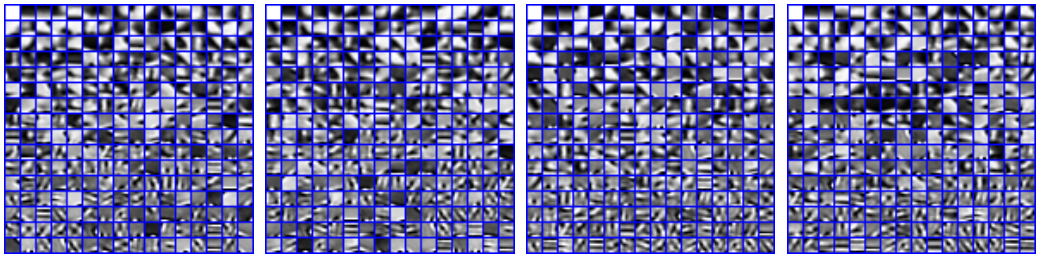
Figure 5.11: Correlation between atom feature and magnitude of nonzero coefficients

having a short zero run length becomes high. This results in more efficient code assignment to zero run length.

Figure 5.13 shows the occurrence probability of nonzero AC coefficient levels for the feature value of atom. Here, the feature value of atom utilized is F_3 which showed the strongest correlation from the measurement results shown in Figure 5.11. After defining $p = int(16 \times F_3/max\,(F_3))$, I measure the probability distribution for each p. This measurement is performed under the condition of $T_0 = 5$ and QP$= 32$. It is clear that the probability distribution is different depending on the feature value of atoms. In addition, Figure 5.14 shows the comparison between occurrence probability of the zero-run length under original order and that after reordering the atoms using the feature value F_3. We can find that the zero-run length has a distribution that concentrates on smaller values for all k compared to before reordering the atoms. Therefore, it was verified that the adaptive code assignment by considering the atom feature is very significant for reducing both the amount of nonzero AC coefficient level and zero run length.



(a) Before reordering (b) Scan order (c) Reordered by $F_1(2)$ (d) Reordered by $F_1(4)$



(e) Reordered by $F_2(2)$ (f) Reordered by $F_2(4)$ (g) Reordered by F_3 (h) Reordered by $F_4(10)$

Figure 5.12: Examples of the atoms reordered by their features

Figure 5.15(a) shows the measured feature value F_3 for each atom in the dictionary designed using K-SVD. The blue line in Figure 5.15 is the result by the conventional zigzag scan order, and the red line is the result by scanning in ascending order of F_3. When arranged in the conventional zigzag scan order, the feature value F_3 fluctuates drastically. As a result, the probability that a coefficient with a large magnitude and a coefficient with a small magnitude will randomly occur becomes higher, and code assignment to zero runs becomes inefficient. If the coefficients are scanned in ascending order of F_3, the probability that coefficients with large magnitude will be concentrated at the beginning of the scan becomes higher, and efficient code assignment can be realized. On the other hand, the measurement results for complete DCT under the same conditions in Figure 5.15(a) are shown in Figure 5.15(b). We can see that even if the proposed method is applied to complete DCT, the scan order is almost unchanged from the zigzag scan used in the conventional method, and the effect of increasing the coding efficiency is small. The reason why the scan order hardly changes even when the proposed method is applied is that zigzag scan itself is already setting effectively for complete DCT whose atom features are already known. Note that the results in Figure 5.15 was confirmed to be the same when not only the feature value F_3 but also other feature values F_1, F_2, and F_4 are used.

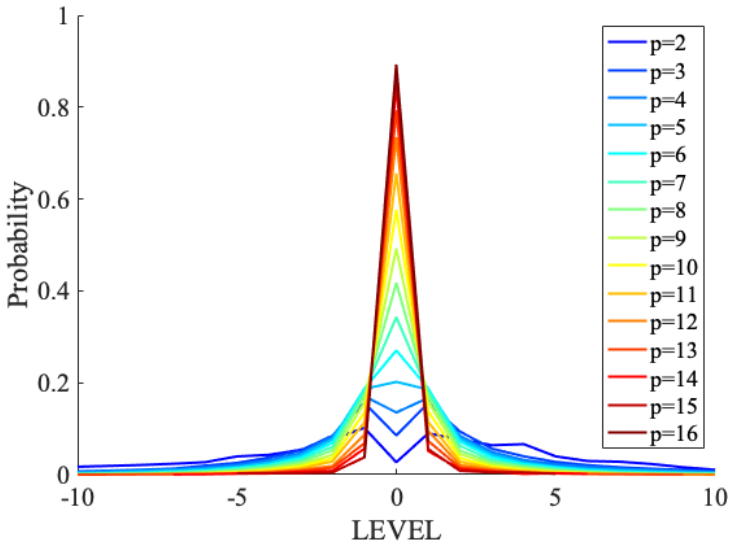


Figure 5.13: Probability distribution of level after reordering

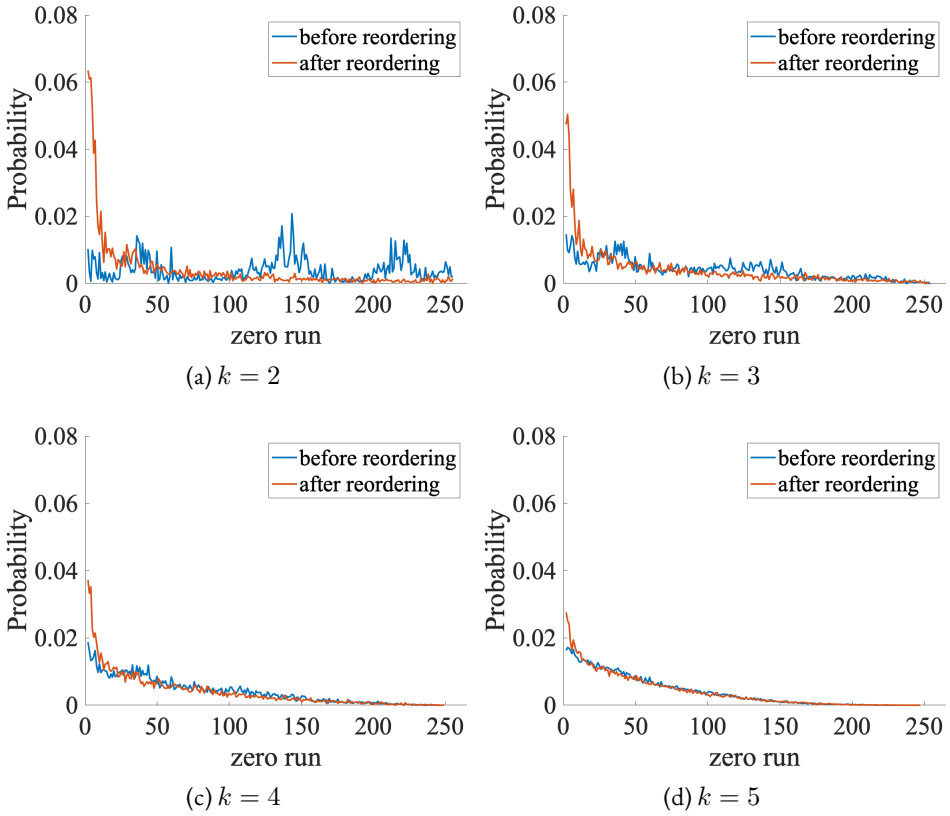
Table 5.1: Simulation conditions

Training data	1.2M 8×8 blocks from [54]
Feature for classifier	DSURF
Number of classes C	16, 32, 64, 128
Initial dictionary	16×16 overcomplete DCT
T_0	3, 5, 7

5.4 Experiments

5.4.1 Experimental conditions

In this section, based on the analysis in section 5.3, simulation experiments are performed under various conditions to verify the coding efficiency. The experimental conditions are shown in Table 5.1. In order to design the dictionary, a total of 1.2 million 8×8 blocks were extracted from the images of the ITE/ARIB HDTV test materials database[54] as training data, and they were classified by DSURF. The multi-class dictionaries were designed under the number of classes of 16, 32, 64 and 128. In each class, a dictionary is designed by K-SVD with an overcomplete DCT of dimension 16×16 (sixteen 8×8 DCT bases in both horizontal and vertical direction) as the initial dictionary. The sparse constraint parameter T_0 was set to 3, 5, and 7.

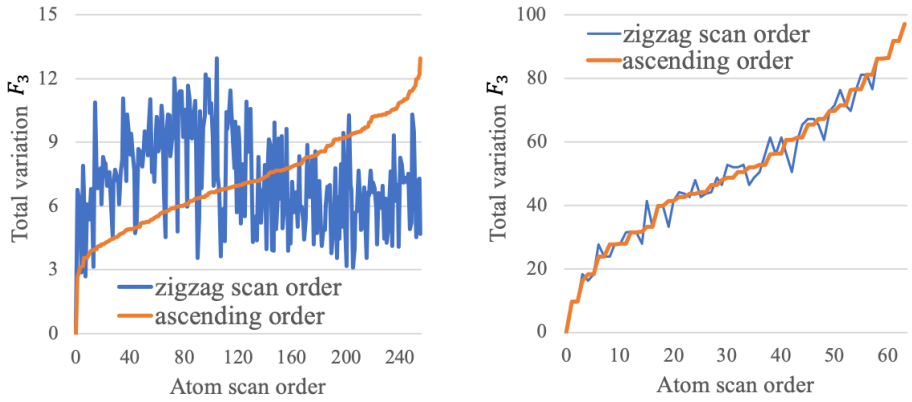


(a) $k = 2$ (b) $k = 3$

(c) $k = 4$ (d) $k = 5$

Figure 5.14: Probability distribution of zero run before and after reordering

If the number of samples used for training is too small, the image representation performance by the designed dictionary will be degraded, and as a result meaningful experiments for this study will not be possible. If the number of samples used for training is large enough and various features of general images are well-balanced in them, the dictionary created by learning will converge to a versatile optimal solution. In this study, the ITE test image database was used for training. This is because it is composed of images with various features targeted for codec evaluation, etc., and it is considered that the features of images that are generally used can be captured sufficiently by using all these images for training. In the conventional studies, training is performed using samples of tens of thousands of blocks (for example, about 68000 blocks in Reference[45] and one hundred thousand blocks in Reference [1]). On the other hand, the number of 1.2 million blocks used in this study is sufficiently large compared to the number of blocks used in the conventional studies. So, it is considered that an appropriate dictionary is designed for entropy coding research, which is the focus of this dissertation.



(a) dictionary designed by K-SVD

(b) complete DCT

Figure 5.15: Probability distribution of zero run before and after reordering

When encoding, first, the image to be encoded is divided into 8×8 small blocks, and their class number are determined by the k-means method according to the DSURF feature of the small block. Next, using OMP and the selected class dictionary, I obtain T_0 sparse coefficients that approximate the small block to be encoded. In OMP, DC atom is always used. Therefore, the number of AC coefficients is $T_0 - 1$. The obtained DC coefficients are quantized with quantization step 1 (i.e., rounded to the nearest integer). On the other hand, AC coefficients are quantized with quantization step QP. In this experiment, I set QP$= 16, 24, 32, 40$. Under these parameters, the effectiveness of introducing zero runs, the effectiveness of adaptation with the number of nonzero coefficients, and the effectiveness of adaptation by feature of atoms feature are examined, in comparison with conventional entropy coding. Note that, in this research, the image quality does not change in case the same QP is used, so the effectiveness can be verified only based on the amount of generated bits. As shown in Section 5.3, the amount of generated information is calculated by the entropy based on the occurrence probability of the symbols to be coded. In the original documents (References[45] and [1]) of the conventional methods to be compared, Huffman codes and Golomb-Rice codes are assigned to the generated symbols. However, for the conventional methods in this experiment, instead of actually assigning a code bit, the amount of information is calculated based on the entropy of the generated symbol in order to make a fair comparison.

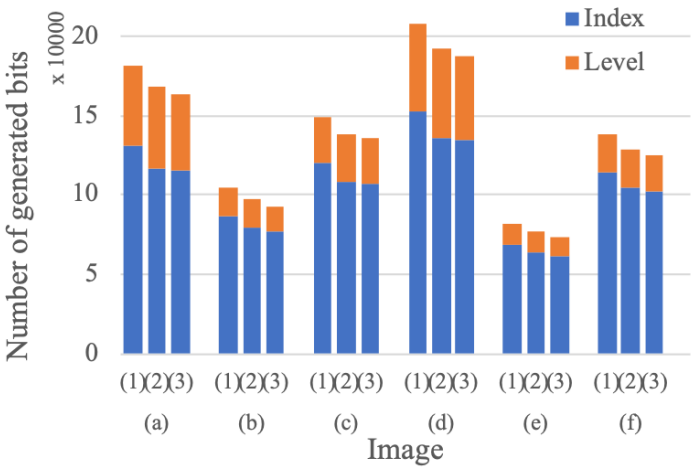


Figure 5.16: Number of generated bits
by (1) conventional[45], (2) conventional(zero run)[1] and (3) k-adaptive, for (a) BQTerrace, (b) BasketballDrive, (c) Cactus, (d) ChristmasTree, (e) Kimono1 and (f) ParkScene. $T_0 = 5$, QP $= 24$

5.4.2 Experimental results

First, under the conditions fixed at $T_0 = 5$ and QP $= 24$, I measured the effectiveness of k-adaptation, i.e. the adaptive encoding by the number of nonzero coefficients. Figure 5.16(1) shows the result of the conventional method based on Reference[45], in which the index of the atom corresponding to the nonzero coefficient is directly encoded.

Figure 5.16(2) shows the result of the conventional method based on Reference [1]), in which the zero-run length between nonzero coefficients is encoded. The k-adaptation is not performed in both Figure 5.16(1) and Figure 5.16(2). On the other hand, Figure 5.16(3) shows the result of applying k-adaptation to Figure 5.16(2). We found from Figure 5.16 that the introduction of zero-run length can reduce the amount of information generated for the indices, so the total amount of information decreases accordingly. However, it should be noted that the amount of information for the quantized level number of the nonzero coefficients has not been reduced. On the other hand, introduction of k-adaptation can not only reduce the amount of information for zero run length but also reduce the amount of information for level number, as a result it is possible to reduce the total amount of information up to 11.0% compared to Reference[45] and up to 4.7% compared to Reference [1]. These characteristics were also found to be similar when using different T_0 and QP.

Next, we verified the effectiveness of adaptation based on the feature of the atoms. The experiment was performed under the condition that the zero-run length and the nonzero

Table 5.2: Number of generated bits (kbit)

(a) conventional[1], (b) k-adaptation, (c) bit saving ratio(%): $\frac{(b)-(a)}{(a)}$,
(d) k-adaptation+atom reordering with the feature F_1, F_2, F_3, F_4,
(e) bit saving ratio(%): $\frac{F_3-(a)}{(a)}$

Images		**(a)**	**(b)**	**(c)**	**(d)**				**(e)**
					F_1	F_2	F_3	F_4	
BQTerrace	Index	117.2	115.8	-1.2	115.7	115.3	115.1	113.8	-1.8
	Level	50.5	47.6	-5.7	46.3	44.0	44.2	45.5	-12.5
	Total	167.7	163.4	-2.5	162.0	159.3	159.2	159.3	-5.0
BasketballDrive	Index	79.8	76.9	-3.6	77.6	73.4	73.6	73.6	-7.7
	Level	17.7	16.0	-9.7	15.1	12.9	12.9	13.4	-27.1
	Total	97.5	92.9	-4.7	92.7	86.4	86.5	87.0	-11.3
Cactus	Index	108.4	106.5	-1.8	104.9	104.0	103.4	103.7	-4.6
	Level	29.9	28.7	-3.8	27.3	24.5	24.7	25.8	-17.5
	Total	138.3	135.2	-2.2	132.2	128.5	128.1	129.5	-7.4
ChristmasTree	Index	136.4	134.8	-1.1	134.2	134.5	134.0	133.7	-1.7
	Level	55.5	52.0	-6.3	51.0	49.6	49.4	50.4	-11.0
	Total	191.9	186.8	-2.6	185.2	184.1	183.5	184.2	-4.4
Kimono1	Index	63.6	61.6	-3.2	60.1	58.6	58.4	59.3	-8.2
	Level	13.1	12.3	-6.2	11.5	9.5	9.6	10.5	-26.9
	Total	76.7	73.9	-3.7	71.5	68.1	68.0	69.8	-11.4
ParkScene	Index	104.6	102.3	-2.2	100.0	99.5	98.8	99.7	-5.5
	Level	24.0	23.0	-4.2	21.7	18.9	18.9	20.3	-21.2
	Total	128.6	125.3	-2.6	121.7	118.4	117.7	120.0	-8.4

AC coefficient level were classified according to the number of nonzero coefficients in each block, and they were encoded independently. A dictionary is created in which the atoms were reordered using the four features defined in section 5.3.2, and compared the amount of bits using the new dictionary with the amount of bits using the original dictionary. The measured results are shown in Table 5.2.

The column (a) in Table 5.2 shows the amount of information generated by the conventional method shown in Reference [1], the column (b) in Table 5.2 shows that generated when k-adaptation is applied to the conventional method, and the column (d) of Table 5.2 shows the amount of generated information when atom reordering is performed in addition to k-adaptation method. The column (c) and (e) in Table 5.2 show the reduction rate

of the amount of generated information for the column (b) and (d) based on the column (a), respectively. Table 5.2 shows that reordering of atoms by any of the four features makes it possible to reduce the amount of generated information compared to before reordering. In particular, it can be confirmed that the amount of generated bits can be minimized when using the atom feature value F_3. The reason is that, the feature value F_3 is highly correlated with the nonzero AC coefficient level as described in section 5.3. As a result, the amount of bits nonzero AC coefficient levels can be reduced by adopting different code assignment rules for them according to the feature of atoms. Also, the reordering of atoms can concentrate the distribution of zero-run length closer to zero, and leads to a reduction the amount of bits for zero-run length.

Finally, I measured the overall performance under setting the feature value used to reorder the atoms to F_3 which is the most effective to reduce the amount of bits. The class number C was set to four types of 16, 32, 64 and 128. For each C, the sparse parameter T_0 was set to 3, 5 and 7, and the quantization parameter QP was set to 16, 24, 32 and 40. The total amount of bits generated was measured as the sum of the amount of bits for the class number, for the number of nonzero coefficients, for coefficients of DC atom and for coefficients of AC atom. The measured average performance gain, BD-rate, between the proposed method and the conventional method is shown in Table 5.3. Table 5.3 shows that the proposed method can reduce the total amount of bits up to 6.2% compared to the conventional method.

K-SVD is a block-based processing similar to DCT-based coding, so the block noise occurs when the compression ratio becomes high. Using the proposed entropy coding method, a smaller quantization step can be used in comparison with the conventional entropy coding methods under the same compression ratio. As a result, block noise can be reduced as shown in Figure 5.17.

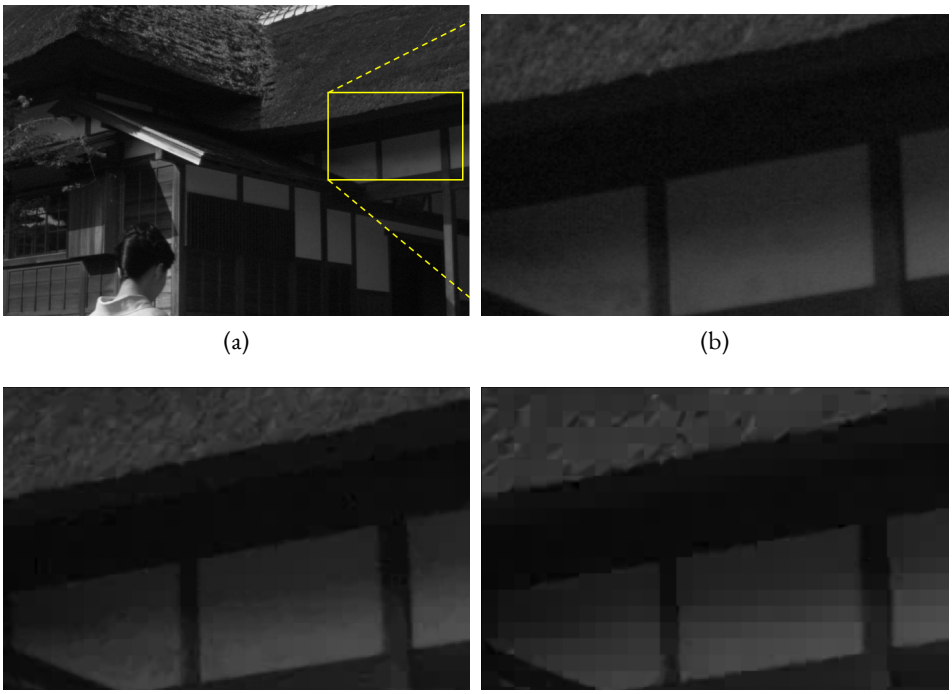
In the experiments by combining the number of classes ($C = 16, 32, 64, 128$), quantization step (QP $= 16, 24, 32, 40$) and sparsity ($T_0 = 3, 5, 7$) as experimental parameters, we clarified that the proposed entropy coding is effective at any bit rate from high compression to low compression. When K-SVD is applied to actual compression coding, multiple parameters of the number of dictionary classes, the quantization step and the sparsity parameter must be controlled in order to keep the amount of generated bits within a predetermined compression ratio. It is considered that the proposed entropy coding method can be utilized for the rate-distortion optimization control for image compression with K-SVD, it will be addressed as a future work.

Table 5.3: BD-rate[%] between proposed method and Reference [1]

Images	T_0	Number of class			
		16	***32***	***64***	***128***
BQTerrace	***3***	-1.43	-1.35	-1.37	-1.31
	5	-5.5	-6.27	-5.56	-6.1
	7	-4.27	-4.21	-4.16	-4.11
BasketballDrive	***3***	-2.03	-2.07	-3.45	-1.33
	5	-0.48	-0.72	-1.08	-0.81
	7	0.44	0.23	0.17	-0.35
Cactus	***3***	-2.37	-3.73	-2.94	-3.67
	5	-3.12	-3.27	-3.53	-3.26
	7	-3.72	-3.36	-3.41	-3.28
ChristmasTree	***3***	-1.13	-0.83	-0.83	-1.06
	5	-1.26	-4.97	-4.27	-1.17
	7	-3.02	-3.0	-3.26	-2.41
Kimono1	***3***	-3.13	-3.08	-2.22	-2.7
	5	-3.0	-2.9	-2.86	-2.61
	7	-3.02	-2.72	-2.62	-2.58
ParkScene	***3***	-4.9	-1.94	-3.87	-3.01
	5	-3.3	-3.8	-3.41	-3.37
	7	-3.53	-3.57	-3.43	-3.5

5.5 Summary

In this chapter, I analyze the statistical properties of nonzero coefficients in detail from theoretical and experimental viewpoints, and propose an efficient entropy coding method of sparse coefficients based on the analysis. Section 5.1 reviews some conventional entropy coding methods and related works of entropy coding. Section 5.2 analyzes the statistical properties of the sparse coefficients in detail. First, I measure the occurrence probability of atom indices and coefficient levels for nonzero coefficients, and clarify the distribution characteristic of zero-run length between nonzero coefficients. Based on the distribution characteristics, in Section 5.3, I propose a context adaptive code assignment method to zero run length and nonzero coefficient level based on the number of nonzero coefficients in the block. Next, I show that the distribution characteristics of nonzero coefficient levels differ depending on features of atoms, and clarify that context adaptive coding to nonzero coeffi-



(a) (b)

(c) (d)

Figure 5.17: Image quality comparison (0.30 bit/pel)
(a) original, (b) enlargement of original image, (c) decoded image (conventional[1]),
(d) decoded image (proposed)

cient levels based on feature of atoms is effective. Furthermore, we found that the zero-run length can be coded efficiently by rearranging the atoms by their features, in Section 5.4. The main focus of this chapter is a research on symbol generation for efficient entropy coding, rather than actual code design method itself such as variable length code tables or arithmetic coding. Therefore, the amount of generated information is discussed mainly based on entropy.

Quantization Method for Sparse Coefficients

6

Contents

The human visual system (HVS) is the best image processing system in the world, but it is far from perfect. the HVS perception of images is non-uniform and non-linear, and does not perceive any variation in the image. For example, image variations caused by quantization errors in image coefficients are not perceptible to the human eye within a certain range. Therefore, if the coding scheme can take advantage of some of the features of HVS, based on which a variety of mathematical models can be built, it is the basis for the development of image digital compression algorithms.

6.1 Human Visual Characteristics

The human eye resembles an optical system, but it is not an optical system in the common sense and is also regulated by the nervous system. The human eye can observe images with the following responses and characteristics.

The human eye's response to changes in the intensity of luminance is nonlinear, and the minimal difference in luminance intensity that is just subjectively discernible by the human eye is usually referred to as the visibility threshold for luminance. That is, when the luminance intensity I increases, it is not felt within a certain range, until the change exceeds a certain value $I + \Delta I$, the human eye can feel the change in luminance, $\Delta I/I$ is also generally known as contrast sensitivity. Therefore, if the error of the recovered image is lower than the contrast sensitivity, it will not be detected by the human eye.

From the viewpoint of the spatial frequency domain, the human eye is a low-pass linear system, because the pupil has a certain geometric size and a certain optical aberration, visual

cells have a certain size, when the spatial plane two black dots close to each other to a certain extent, the observer away from the black dots at a distance cannot distinguish between them, which means that the human eye's ability to distinguish the details of the scene is limited, this limit is the resolution. Studies have shown that the resolution of the human eye has the following characteristics.

1. The human eye resolution decreases when the illumination is too strong or too weak or when the background luminance is too strong.

2. The resolution of the human eye decreases when the speed of visual target motion increases.

3. The resolution of the human eye is worse for color details than for luminance details; if the black-and-white resolution is 1, it is 0.4 for black-red and 0.19 for green-blue.

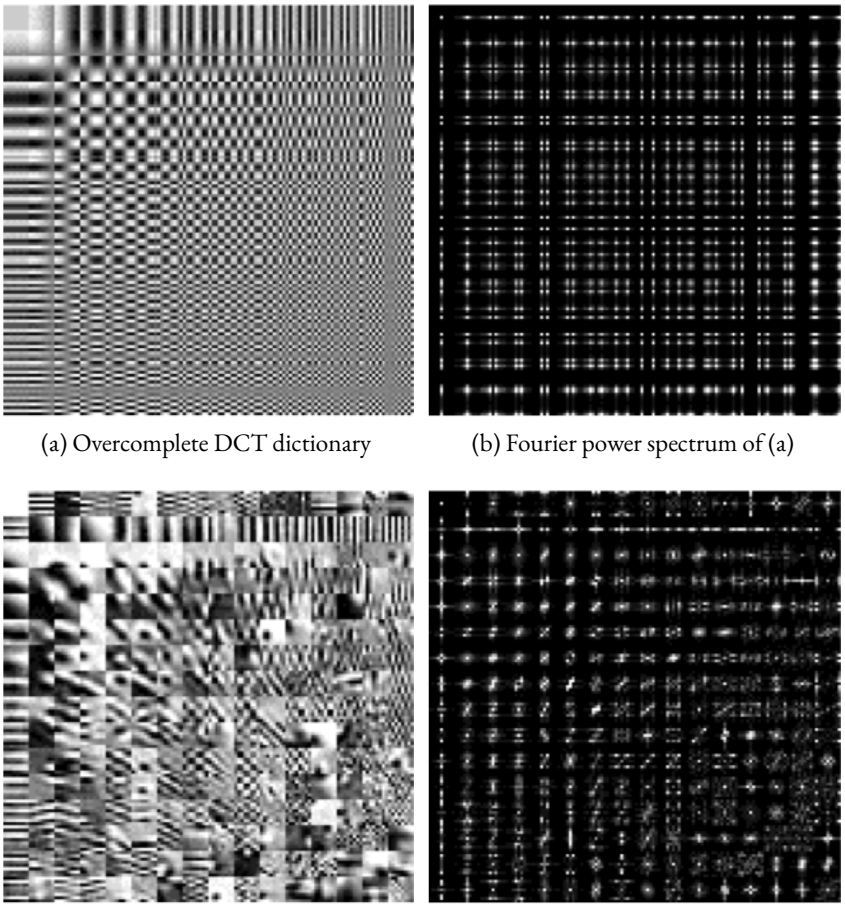
In addition, HVS is insensitive to high frequency signals.

These characteristics of HSV are important for the design of image coding quantizers, and exploiting these visual characteristics allows images to tolerate larger quantization errors, which can lead to reduced quantization levels and thus lower bit rates.

6.2 Frequency Characteristics of Atoms

In this section, we discuss the frequency characteristics of the atoms contained in the dictionary that enable sparse representations. For easy understanding, the element of k-th atom d_k is expressed as $d_k(p, q)$ according to the $M \times M$ two-dimensional input block, and the element of coefficient vector $\boldsymbol{x}$ is expressed as $\boldsymbol{x}(k)$. Here, $1 \leq p, q \leq M$. The variance of $d_k(p, q)$ is normalized to 1. As mentioned above, the human eye is sensitive to changes in low-frequency signals, but is not good at distinguishing changes in high-frequency signals. Using this fact, JPEG or HEVC perform fine quantization on the coefficients for the low frequency atoms of DCT and coarse quantization for the coefficients on the high frequency atoms of DCT.

Figure 6.1(a) shows the atoms of the overcomplete DCT when the block size is 8×8, and Figure 6.1(b) shows the distribution of the power spectrum when the digital Fourier transform (DFT) is performed to each atom. The center of the power spectrum in Figure 6.1(b) shows the DC component. This analysis shows that the frequency characteristics of the atoms in overcomplete DCT are regularly arranged. On the other hand, Figure 6.1(c) is an example of a dictionary that enables sparse representation. In addition, Figure 6.1(d)



(a) Overcomplete DCT dictionary

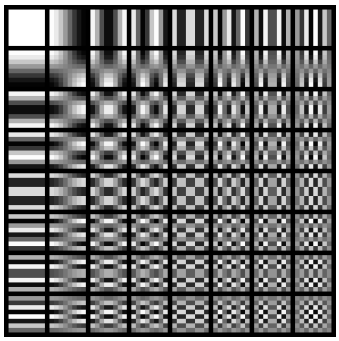
(b) Fourier power spectrum of (a)

(c) A Dictionary designed by K-SVD

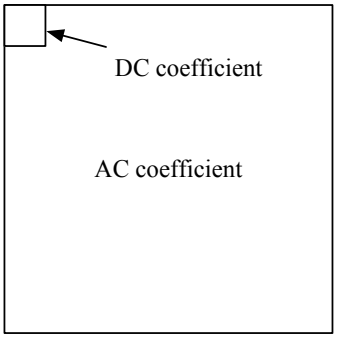
(d) Fourier power spectrum of (c)

Figure 6.1: Dictionary atoms and their Fourier power spectrum

shows the power spectrum obtained by DFT of each atom in Figure 6.1(c). The shape of the atoms included in this dictionary reflects the characteristics of the blocks included in the class used to design the dictionary, and it is found that their frequency characteristics are much more complicated and have more variation than the frequency characteristics of overcomplete DCT. Therefore, it is important to consider how to control the quantization width of the weighting coefficients for each atom.



16	16	16	16	17	18	21	24
16	16	16	16	17	19	22	25
16	16	17	18	20	22	25	29
16	16	18	21	24	27	31	36
17	17	20	24	30	35	41	47
18	19	22	27	35	44	54	65
21	22	25	31	41	54	70	88
24	25	29	36	47	65	88	115



(a) (b) (c)

Figure 6.2: 8×8 DCT atoms and quantization matrix in HEVC

6.3 Quantization Matrix Design

In HEVC[58], the default 8×8 intra quantization matrix for integer DCT are derived from the modulation transfer function (MTF) model[63]. Figure 6.2 shows an example of the 8×8 DCT atoms in HEVC and the corresponding Q-matrix for intra coding. This means that the quantization step width of the AC coefficient is $Q_{ratio}(\equiv w(k)16)$ times that of the DC coefficient, where $w(k)$ is the element of the matrix. Since the atoms of DCT are regularly expressed in frequency, the Q-matrix can be theoretically determined. However, many of the atoms in the dictionary for sparse representation are very complicated, and there has been no way to determine an appropriate Q-matrix. Therefore, in this paper, a quantization matrix is designed by simply matching the complexity of the sparse representation atoms with the complexity of the DCT atoms.

First, for each DCT atom, the complexity "R" is calculated. The total variation defined by

$$\begin{aligned} R(k) &= \sum_{p} \left| \frac{\partial}{\partial p} d_k(p,q) \right| + \sum_{q} \left| \frac{\partial}{\partial p} d_k(p,q) \right| \\ &= \sum_{p=1}^{M-1} \sum_{q=1}^{M} |d_k(p+1,q) - d_k(p,q)| \\ &+ \sum_{p=1}^{M} \sum_{q=1}^{M-1} |d_k(p,q+1) - d_k(p,q)| \end{aligned} \tag{6.1}$$

is used for complexity calculation. Here, $d_k(p,q)$ means k-th atom. As illustrated in Figure 6.3 the total variation is defined as the sum of the absolute value of the difference be-

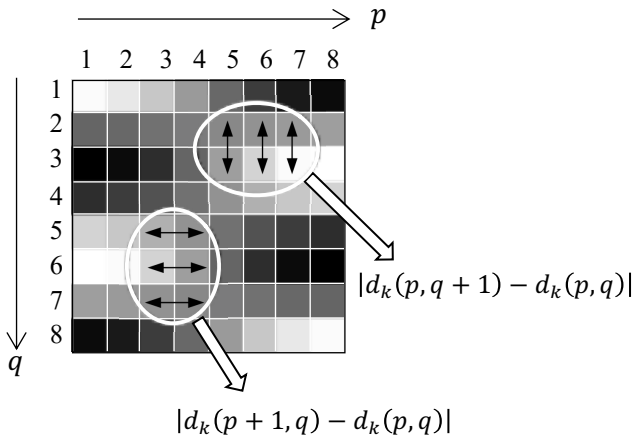


Figure 6.3: Total variation of an atom

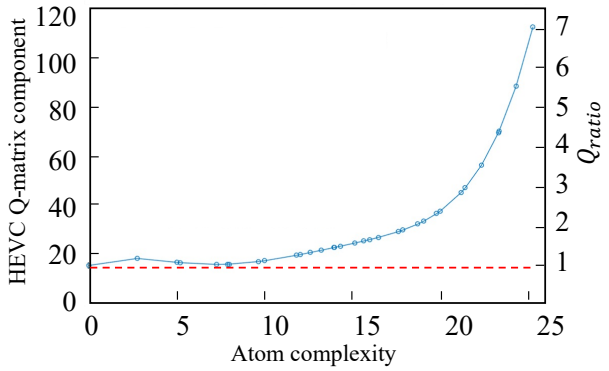


Figure 6.4: Relationship between atom complexity and Q-matrix component

tween the neighboring pixels in the horizontal and vertical direction within a block. The relationship between $R(k)$ and the corresponding quantization matrix value $w(k)$ is plotted as shown in Figure 6.4. This gives an approximate curve $w(k) = f\Big(R(k)\Big)$.

Next, the same complexity is computed as in Equation (6.1) for each atom in the sparse representation dictionary. Figure 6.5 shows some of the atoms in the sparse representation dictionary and the complexity R of those atoms calculated by Equation (6.1). Clearly, R reflects the complexity of the frequency characteristics of the atom.

Finally, by mapping the complexity on the graph of Figure 6.4, we obtain the Q-matrix components for each atom. Figure 6.6 shows two examples of Q-matrices for sparse representation dictionaries based on the described procedure.

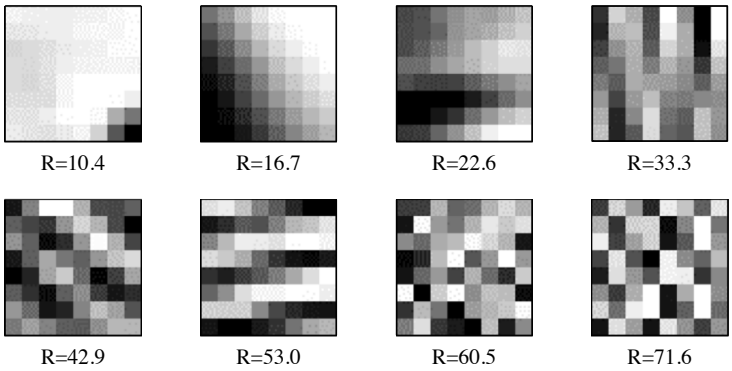
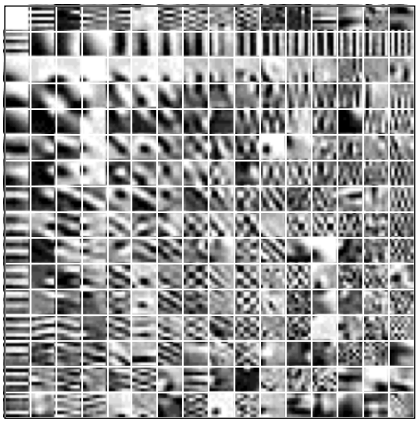


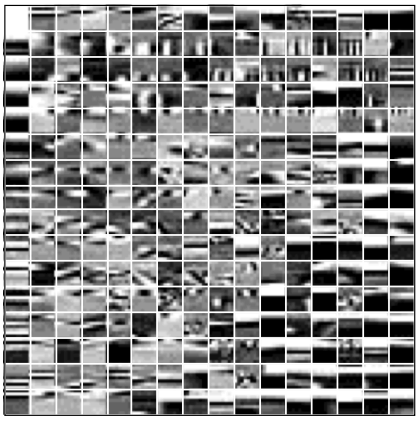
Figure 6.5: Atom examples and their complexity R



(a) Dictionary: class ♯1

15 115 17 77 23 29 60 32 29 16 84 30 36 34 30 21
24 17 16 16 17 17 17 17 17 17 17 17 17 18 18 17
17 17 16 17 16 16 17 16 17 17 16 17 18 17 17 16
16 17 16 17 16 16 17 16 16 16 18 16 17 18 15 16
16 16 18 17 16 16 17 16 18 17 16 16 17 17 16 15
15 15 16 15 15 16 15 16 16 17 15 16 15 18 18 16
16 15 16 16 15 16 15 16 15 17 18 17 17 16 16 18
21 15 19 15 18 15 17 16 17 16 15 18 17 17 18 23
18 16 17 19 18 19 18 22 18 24 17 22 19 22 22 20
21 20 25 15 17 21 18 18 16 15 15 19 21 23 15 32
27 29 23 17 23 24 21 26 17 22 16 21 28 25 24 16
25 39 26 15 29 20 22 27 16 15 17 16 15 28 24 26
63 30 27 26 31 16 15 34 25 50 24 21 29 28 35 27
41 33 30 18 32 20 41 63 19 18 22 15 23 25 16 35
67 17 17 30 58 16 40 26 18 26 29 20 27 24 31 32
71 15 31 68 29 17 36 21 24 17 17 15 16 26 20 30

(b) Q-matrix for class ♯1



(c) Dictionary: class ♯2

17 32 24 21 25 20 30 16 19 16 18 27 20 24 15 16
20 16 16 16 16 15 15 15 15 18 16 15 15 15 15 16
16 16 15 16 19 16 16 15 15 15 15 16 20 16 16 35
16 18 22 17 21 15 15 22 24 15 17 26 16 16 16 20
16 15 15 16 15 15 15 15 16 15 15 15 15 15 15 15
19 20 16 18 16 18 19 18 19 20 19 22 32 21 19 17
16 23 19 19 18 17 20 21 20 19 16 32 27 16 16 15
17 22 17 16 18 23 20 15 18 32 16 20 20 24 15 16
23 23 23 23 19 19 18 20 21 24 21 21 25 25 17 21
27 18 20 18 18 25 20 35 21 22 24 16 24 17 15 18
28 18 25 26 28 26 19 23 26 19 23 18 17 19 19 16
34 22 19 19 26 18 22 19 16 17 15 16 16 23 18 17
54 20 25 28 22 18 16 28 22 18 16 21 17 16 16 17
25 22 17 20 16 17 16 21 20 21 16 26 17 19 21 18
76 41 24 28 34 29 22 16 21 23 15 22 17 20 16 20
26 18 19 17 19 22 16 16 26 20 18 16 16 16 18 18

(d) Q-matrix for class ♯2

Figure 6.6: Examples of the designed Q-matrix

Table 6.1: Experimental conditions

Process		Parameters / Conditions
Dictionary training	Block size	8×8
	Classification	128 classes using dense SURF
	Training samples	2688 images with 256×256
	Designed dictionaries	Sparsity: $T_0 = 3$ Number of classes: $C = 128$ Number of atoms in each: 16×16 overcomplete atoms
Coding	Test images	"Lighthouse", "Yacht" and "Sea"
	Quantization	Midtread-type linear quantization
	Quantization matrix	Proposed (total variation based) and conventional (uniform)
Evaluation	Picture quality	PSNR, SSIM, MOS (Absolute Category Rating)

6.4 Experiments

6.4.1 Experimental conditions

Experimental conditions are shown in Table 6.1. For training the dictionary, I used 2688 images of size 256×256 and generated the dictionaries consisting of 128 classes based on dense SURF. The block size is set to 8×8. The dictionary for each class was generated from an overcomplete DCT consisting of 16×16 atoms under the sparsity $T_0 = 3$. Three images, "Lighthouse", "Yacht", and "Sea" in the Kodak dataset are used as test data. None of them belong to the training dataset.

In this experiment, the sparse coefficient $\boldsymbol{x}(k)$ is linearly quantized to $\boldsymbol{x}_Q(k)$ by

$$\boldsymbol{x}_Q(k) = \mathrm{sign}\Big(\boldsymbol{x}(k)\Big) \times \lfloor \frac{|\boldsymbol{x}(k)| + \mathrm{QP}'/2}{\mathrm{QP}'} \rfloor \tag{6.2}$$

$$\mathrm{QP}' = \mathrm{QP} \times w(k)/16 \tag{6.3}$$

Here, QP is a quantization parameter to control the amount of generated bits, and $w(k)$ is the Q-matrix. The amount of generated information was calculated by the entropy based

Table 6.2: MOS scores on ACR

Rating	Image quality
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

on the probability distribution of the zero-run distance between the nonzero coefficient positions and the probability distribution of the quantization level values of the nonzero coefficients. Also, since there are 128 classes, the class number information is 7-bit fixed length coding. Regarding the image quality evaluation, I performed an objective evaluation by PSNR and SSIM, and a subjective evaluation based on the ACR (Absolute Category Rating) method[64]. Table 6.2 shows the relationship between image quality levels and corresponding scores in subjective evaluation experiments. The number of examinees for the subjective evaluation is 16.

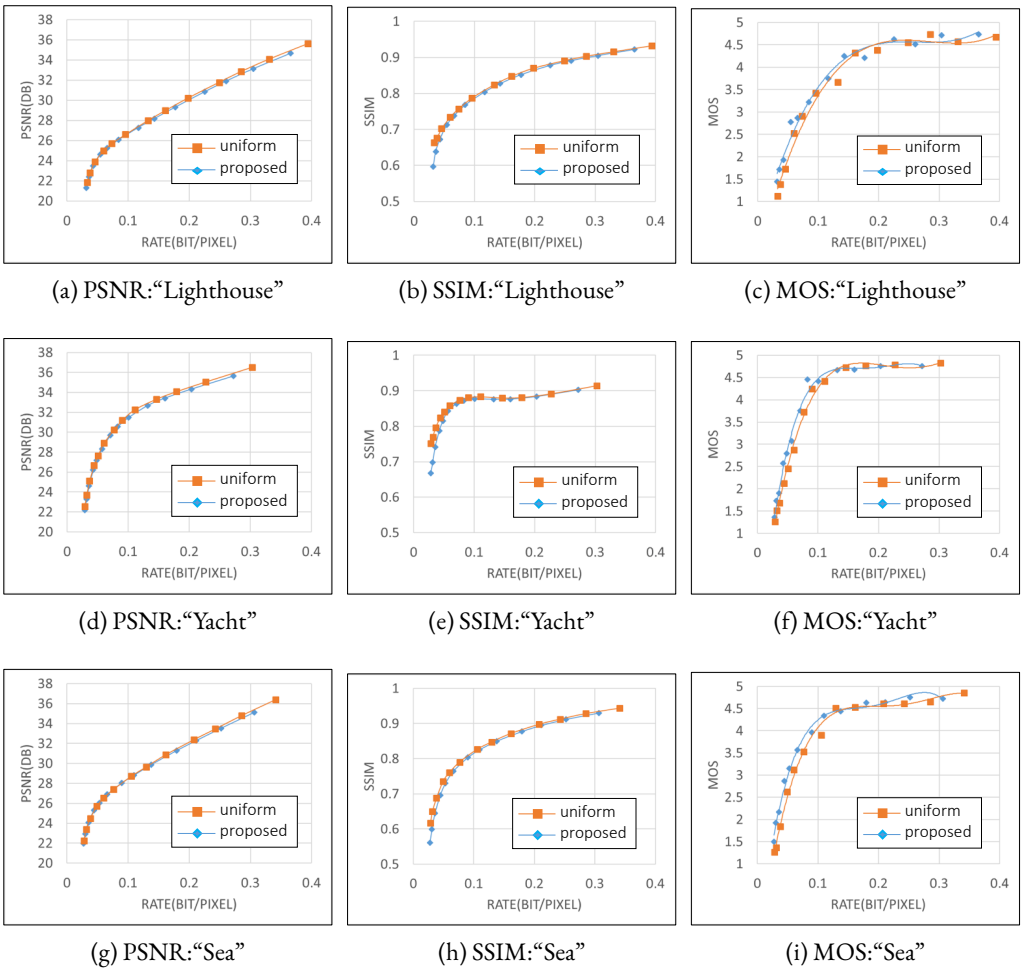
6.4.2 Experimental results

Figure 6.7 shows RD characteristics comparison. The coding distortion was compared with PSNR and SSIM measured for the decoded image, and the Mean opinion score (MOS) obtained by analyzing the results of the subjective evaluation experiment. In Figure 6.7, the evaluation values when the sparse coefficients are quantized by the proposed quantization matrix are shown in comparison with the evaluation values when all the coefficients are quantized by the uniform quantization matrix. From these results, we can find that the proposed method is slightly worse for PSNR and SSIM, but the proposed method is superior from the viewpoint of subjective image quality.

The reason that the PSNR of the proposed quantization matrix is lower than the PSNR of the uniform quantization matrix is that the proposed method coarsely quantizes the complicated edge regions including high frequency components. The change of pixel value in the complicated edge regions greatly contributes to the square error used in the PSNR calculation. On the other hand, SSIM is generally considered as an objective evaluation standard that reflects perceptual impressions.

However, in this experiment, the SSIM of the proposed method is inferior to SSIM of the conventional method using a uniform quantization matrix, especially at lower bit rates.

When a uniform quantization matrix is used, the quantization width becomes relatively coarse for the gradation regions where the brightness changes smoothly, and a largely noticeable false contours occurs. However, SSIM cannot reflect the subjective image quality degradation associated with such severe false contouring. This is the reason why the SSIM of the conventional method is not lower than the SSIM of the proposed method.



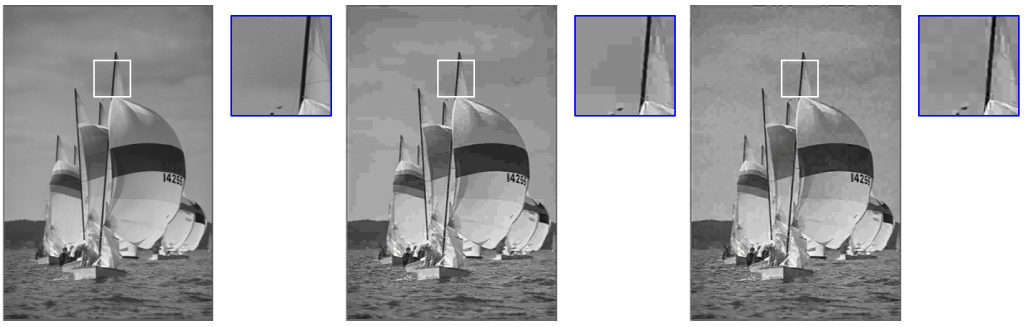
(a) PSNR:“Lighthouse” (b) SSIM:“Lighthouse” (c) MOS:“Lighthouse”

(d) PSNR:“Yacht” (e) SSIM:“Yacht” (f) MOS:“Yacht”

(g) PSNR:“Sea” (h) SSIM:“Sea” (i) MOS:“Sea”

Figure 6.7: Picture quality comparison
Top row: “Lighthouse”, middle row: “Yacht”, bottom row: “Sea”,
left column: PSNR, middle column: SSIM, right column: MOS

(a) original:“Lighthouse” (b) uniform Q-matrix (c) proposed Q-matrix



(d) original:“Yacht” (e) uniform Q-matrix (f) proposed Q-matrix

(g) original:“Sea” (h) uniform Q-matrix (i) proposed Q-matrix

Figure 6.8: Comparison of the decoded image under the same bitrate (0.05 bpp)

Figure 6.8 shows the decoded images under the same compression ratio. From the viewpoint of subjective image quality, it is clear that the perceptually noticeable false contours occurred in the regions having gradually changing luminance value by the conventional quantization. On the other hand, the method using the proposed quantization matrix can reproduce the luminance change close to the original image. The effectiveness of the proposed quantization matrix was verified because MOS is improved compared to MOS of the conventional method, especially at medium and low bit rate.

6.5 Summary

In this chapter, I have proposed a new quantization matrix design method using total variation of atoms for sparse representation-based image coding.

In section 6.1, the characteristics of the Human Visual System (HVS) is introduced, and frequency characteristics of atoms in sparse dictionary were analyzed in section 6.2.

Followed by section 6.3, a method based on the results of analyzation, for quantizing sparse coefficients using Q-Matrix is proposed.

In section 6.4, experimental results show that subjective image quality provides a MOS that is 0.2 to 0.3 points higher than conventional uniform quantization, and the proposed method can provide higher coding efficiency from a point of perceptual picture quality.

Conclusions 7

Contents

7.1 Summary

Even today, when the communication speed and environment are constantly upgrading and improving, image compression remains an issue that cannot be ignored in the face of the rapid growth of the visual information explosion. Among these issues, transform coding, as an indispensable part of image compression solutions, has been attracting the interest of researchers. Transform dictionaries such as DCT, which are designed based on statistical perspectives, are difficult to satisfy the optimization of transformation efficiency for a particular image. To cope with these problems, this dissertation attempts to develop new image compression methods by importing sparse coding dictionary generation algorithms into transform coding to further improve the efficiency of image compression.

For the procedure of transform coding, we note that the current techniques and literature can be divided into two categories. One of which is to generate a transform dictionary by structurally decomposing the assumed data distribution model, such as Fourier transform, discrete cosine transform, wavelet transform. This type of algorithm has an advantage in terms of computational capacity due to the existence of structured modeling, and is able to achieve high speed computation of results with relative ease. In contrast, the idea of generating dictionaries from real data through machine learning, including maximum likelihood, MOD, PCA, and K-SVD, is relatively complex. However it can be more adapted to the structure of the target data itself than the dictionaries generated by the former category, rather than an abstract predictive model. We believe that the latter has the potential to further improve the efficiency of existing transform coding.

Sparse representation for images

Among the learning dictionary generation algorithms in recent years, the K-SVD algorithm is often mentioned as a milestone which draws on the design ideas of the classical classification algorithm k-means and greatly optimizes the computational speed of the algorithm for dictionary generation. K-SVD algorithm is used to generate sparse dictionaries. We analyzed the performance of dictionaries generated for each image, a single dictionary generated from multiple images, and a multi-class dictionary generated from multiple images. In the first approach, the designed dictionaries are overly dependent on the original image, making it difficult to apply them to other images. The single dictionary generated by the second approach converges to a statistically optimal average model, i.e., a DCT-like feature. The dictionary is too versatile to code efficiently for a specific image. From a viewpoint of efficient representation of various images, the third approach is the most balanced method.

Dictionary design based on Multi-class K-SVD with iterative class update

A common problem with some existing proposals for generating multi-class dictionary from many images is that the classification result and the performance of the final generated dictionary are independent and delinked. In other words, if the performance of the final dictionary generated is impacted by the classification results, it is difficult for existing algorithms to actively avoid this pitfall. To address this problem, I proposed a dictionary design algorithm with class iterative update function. And I experimentally compared the performance of this algorithm with the one without class iteration to prove the effectiveness of the proposal.

Entropy coding method for sparse coefficients

The coefficients generated by the sparse dictionary have very different characteristics from those of the coefficients encoded by the traditional transformation encoding. In other words, it is not an appropriate idea to apply the traditional coefficient encoding directly to the sparse coefficients. This dissertation discussed this problem. I first analyzed the statistical characteristics of the sparse coefficients and the distribution patterns of the valid data (non-zero values), and then designed and proposed an adaptive entropy coding algorithm according to the coefficient characteristics. Its contribution to the coding performance improvement is experimentally confirmed.

Quantization method for sparse coefficients

In addition to entropy coding, another important component of the whole procedure of image compression is quantization. The main optimization of quantization parameters for the transform coefficients in international standardization of image compression is to use a technique called Quantization Matrix (Q-Matrix), to quantize the coefficients for different atoms with different quantization steps. This technique takes into account the fact that the human visual system is insensitive to change of high frequency signals in the spatial domain and sensitive to change of low frequency signals. That is, the quantization step width of the coefficients corresponding to atoms with different frequency features varies based on quantization matrix values. Quantization matrices for the coefficients corresponding to atoms with regular frequency characteristics such as DCT have been studied extensively and are optimally designed from a point of HVS. However, the frequency characteristics of the atoms contained in the sparsely expressible dictionary obtained by K-SVD are extremely complicated, and it was not clear what kind of quantization step width should be used to quantize a coefficient corresponding to each atom. In this dissertation, a new quantization matrix design method for sparse coefficients was shown by quantifying the spatial frequency features of each basis contained in a dictionary that enables sparse representation. And it was experimentally verified that this Q-Matrix design method plays a positive role in enhancing the subjective image quality of the compressed image.

7.2 Future Directions

Prediction residual

The training samples used in existing work are taken from image data, where it is well considered that there is a large amount of visual signal redundancy in image data, and machine learning is able to abstract them into dictionary atoms with high representational capability. We also note that in existing efficient video compression techniques, the main transform coding object is the predicted residuals, because the redundancy of the self-similarity of the image can be eliminated in advance through prediction, and the residuals are further squeezed in the transform coding process to achieve high compression efficiency.

It should be noted here that the fact which the residual data can be further compressed by transform coding is a proof that some structural similarity among the data is not deleted cleanly in the prediction process, or a portion of the residual data with similar structure is generated.

Nevertheless, it can be predicted that in order to effectively generate a usable dictionary from the residual data by machine learning means, a much larger amount of training data than that of the real data is required. And, another feasible approach is to perform some special processing on the residual data in the data pre-processing stage to reduce the learning difficulties caused by the sparsity of the residual data. Learning a sparse representation dictionary from the intrinsically sparse data is a challenging topic in the field of machine learning.

Application to Videos

The main object of this work is still image compression, so a natural extension of this direction is to apply it to the field of video compression. Video has an additional dimension - time - than static graphs, so naturally there is redundancy in the time direction, and current international standards have proposed motion compensation prediction techniques for such redundancy. Similar to the directions mentioned in the previous subsection, extending the proposal of this dissertation to the residuals of motion-compensated prediction is also a research direction worth discussing.

Switching strategy under RD optimization

The dictionary proposed in this dissertation involves the selection of many parameters when it is generated and applied to image compression. After comparing the experimental results, I have proposed the globally optimal recommended values within the scope of the experimental parameters. However, if these parameters can be dynamically adjusted according to the data they correspond to, i.e., it is possible to dynamically switch between dictionaries generated with different parameters, which will offer the possibility to further improve the compression efficiency.

Class merge / atoms rearrange module

In the dictionary update stage, we made adjustments to the data classification based on the performance of the dictionary generated in the previous iteration. During the adjustment process, we noted some classes whose usage may have been affected in some way. They may have been learned from uncommon image blocks (not often used), or there may have been some similarity between block features that several classes are good at representing (thinned out in usage ratio by other classes). This leads to an interesting direction of re-

search - whether a module could be added to the update process to determine the similarity between dictionaries of different groups, similar to the coefficient of determination[65] or correlation coefficient[66] as in regression analysis, to dynamically plan whether the existing number of classes is appropriate or not.

Moreover, once a decision is made to merge some classes, the atoms within them need to be rearranged or eliminated to some extent, similar to the pruning process in deep learning, so this rearrangement strategy is worth exploring.

Reference

[1] R. Vinith, A. Aswani, and K. Govindan, "Medical Image Compression Using Sparse Approximation," *International Journal of Advanced Computer and Mathematical Sciences*, vol. 6, no. 2, pp. 30–39, Jul. 2015.

[2] ITU-R, "BT.2020: Parameter values for ultra-high definition television systems for production and international programme exchange," International Telecommunication Union (ITU-R), Tech. Rep., Aug. 2012, https://www.itu.int/rec/R-REC-BT.2020/en.

[3] Ichigaya, Atsuro, Nishida, and Yukihiro, "Required Bit Rates Analysis for a New Broadcasting Service Using HEVC/H.265." *IEEE Transactions on Broadcasting*, 2016, http://dx.doi.org/10.1109/tbc.2016.2550778.

[4] "Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper," https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html, 2018.

[5] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, vol. 2, Nov. 2003, pp. 1398–1402.

[6] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *Acm Transactions on Graphics*, 2004, http://www.researchgate.net/publication/2896183_Colorization_using_Optimization.

[7] K. Uruma, K. Konishi, T. Takahashi, and T. Furukawa, "Colorization-based image coding using graph Fourier transform," *Signal Processing Image Communication*, 2018, http://www.onacademic.com/detail/journal_1000041587758699_0c39.html.

[8] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '02*. San Antonio, Texas: ACM Press, 2002, p. 277, http://portal.acm.org/citation.cfm?doid=566570.566576.

[9] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be color! : Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification," *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 1–11, Jul. 2016, https://dl.acm.org/doi/10.1145/2897824.2925974.

[10] Y. Matsuo, S. Iwamura, K. Iguchi, and S. Sakaida, "Real-time Encoding System for Ultra High-definition Video Using Super-resolution Technique," *The Journal of the Institute of Image Information and Television Engineers*, vol. 70, no. 1, pp. J22–J28, 2016, https://www.jstage.jst.go.jp/article/itej/70/1/70_J22/_article/-char/ja/.

[11] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: Image and video synthesis using graph cuts," in *ACM SIGGRAPH 2003 Papers on - SIGGRAPH '03*. San Diego, California: ACM Press, 2003, p. 277, http://portal.acm.org/citation.cfm?doid=1201775.882264.

[12] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, "Overview of SHVC: Scalable Extensions of the High Efficiency Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 20–34, Jan. 2016, http://ieeexplore.ieee.org/document/7172510/.

[13] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," in *ACM SIGGRAPH 2007 Papers on - SIGGRAPH '07*. San Diego, California: ACM Press, 2007, p. 10, http://portal.acm.org/citation.cfm?doid=1275808.1276390.

[14] X. Yu, Z. Liu, J. Liu, Y. Gao, and D. Wang, "VLSI friendly fast CU/PU mode decision for HEVC intra encoding: Leveraging convolution neural network," in *2015 IEEE International Conference on Image Processing (ICIP)*. Quebec City, QC, Canada: IEEE, Sep. 2015, pp. 1285–1289, http://ieeexplore.ieee.org/document/7351007/.

[15] J. Li, B. Li, J. Xu, R. Xiong, and W. Gao, "Fully Connected Network-Based Intra Prediction for Image Coding," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3236–3247, Jul. 2018, https://ieeexplore.ieee.org/document/8319436/.

[16] S. Jimbo, J. Wang, and Y. Yashima, "Deep Learning-based Transformation Matrix Estimation for Bidirectional Interframe Prediction," in *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*. Nara, Japan: IEEE, Oct. 2018, pp. 726–730, https://ieeexplore.ieee.org/document/8574691/.

[17] O. Bryt and M. Elad, "Compression of facial images using the K-SVD algorithm," *Journal of Visual Communication and Image Representation*, vol. 19, no. 4, pp. 270–282, May 2008, http://linkinghub.elsevier.com/retrieve/pii/S1047320308000254.

[18] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[19] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, Nov. 1993, pp. 1–5, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=342465.

[20] S. Takamura, "Evolutionary Video Coding," *The Journal of the Institute of Image Information and Television Engineers*, vol. 69, no. 2, pp. J38–J44, 2015, https://www.jstage.jst.go.jp/article/itej/69/2/69_J38/_article/-char/ja/.

[21] A. M. Kondoz, *Digital Speech: Coding for Low Bit Rate Communication Systems*, 2nd ed. Chichester: Wiley, 2004.

[22] Phil Schniter, *An Introduction to Source-Coding: Quantization, DPCM, Transform Coding, and Sub-Band Coding*, 2nd ed. OpenStax CNX, Sep. 2009, http://cnx.org/contents/b6387028-5ef5-4dbc-a7eb-ee9ea96d1b94@2.1.

[23] A. Habibi, "Hybrid Coding of Pictorial Data," *IEEE Transactions on Communications*, vol. 22, no. 5, pp. 614–624, May 1974.

[24] ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, "Advanced Video Coding for Generic Audiovisual Services," https://www.itu.int/rec/T-REC-H.264-200305-S/en, May 2003.

[25] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[26] "Developing a video compression algorithm with capabilities beyond HEVC," https://www.itu.int/en/ITU-T/studygroups/2017-2020/16/Pages/video/jvet.aspx.

[27] K. R. Rao and P. C. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Boston: Academic Press, 1990.

[28] *A Wavelet Tour of Signal Processing*. Elsevier, 2009, https://linkinghub.elsevier.com/retrieve/pii/B9780123743701X00018.

[29] J. J. Gerbrands, "On the relationships between SVD, KLT and PCA," *Pattern Recognition*, vol. 14, no. 1, pp. 375–381, Jan. 1981, http://www.sciencedirect.com/science/article/pii/0031320381900820.

[30] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Upper Saddle River, N.J: Prentice Hall, 1999.

[31] "Official JPEG website," https://jpeg.org/.

[32] International Telecommunication Union, "T.81: Information technology - Digital compression and coding of continuous-tone still images - Requirements and guidelines," https://www.itu.int/rec/T-REC-T.81/en, Sep. 1992.

[33] International Telecommunications Union -Telecommunication Standardization Sector (ITU-T), "Run-Length Colour Encoding Series T: Terminals For Telematic Services," https://www.itu.int/rec/T-REC-T.45-200002-I/en, Feb. 2000.

[34] D. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952, http://ieeexplore.ieee.org/document/4051119/.

[35] P. Howard and J. Vitter, "Arithmetic coding for data compression," *Proceedings of the IEEE*, vol. 82, no. 6, pp. 857–865, Jun. 1994.

[36] "JCT-VC - Joint Collaborative Team on Video Coding," https://www.itu.int/en/ITU-T/studygroups/2017-2020/16/Pages/video/jctvc.aspx.

[37] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004, http://ieeexplore.ieee.org/document/1284395/.

[38] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," VCEG Meeting (ITU-T SG16 Q.6), Austin, Texas, USA, Tech. Rep., 2001.

[39] T. K. Tan, R. Weerakkody, M. Mrak, N. Ramzan, V. Baroncini, J.-R. Ohm, and G. J. Sullivan, "Video Quality Evaluation Methodology and Verification Testing of HEVC Compression Performance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 76–90, Jan. 2016, http://ieeexplore.ieee.org/document/7254155/.

[40] S. Pateux, "Tools for proposal evaluations," Joint Collaborative Team on Video Coding (JCT-VC), Germany, Tech. Rep. JCTVC-A031, 15, https://www.itu.int/wftp3/av-arch/jctvc-site/2010_04_A_Dresden/.

[41] Bruno A. Olshausen and David J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision Research*, vol. 37, no. 23, pp. 3311–3325, Dec. 1997, https://www.sciencedirect.com/science/article/pii/S0042698997001697.

[42] K. Engan, S. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, vol. 5, Mar. 1999, pp. 2443–2446 vol.5.

[43] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 1, Jun. 2003, pp. I–I.

[44] Jens-Rainer Ohm, Gary J. Sullivan, Heiko Schwarz, Thiow Keng Tan, and Thomas Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012, http://ieeexplore.ieee.org/document/6317156/.

[45] Je-Won Kang, M. Gabbouj, and C.C. Jay Kuo, "Sparse/DCT (S/DCT) Two-Layered Representation of Prediction Residuals for Video Coding," *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2711–2722, Jul. 2013, http://ieeexplore.ieee.org/document/6494295/.

[46] Je-Won Kang, C.C. Jay Kuo, Robert Cohen, and Anthony Vetro, "Efficient dictionary based video coding with reduced side information," in *IEEE International Symposium of Circuits and Systems (ISCAS)*. IEEE, May 2011, pp. 109–112, http://ieeexplore.ieee.org/document/5937513/.

[47] R. Walha, F. Drira, F. Lebourgeois, C. Garcia, and A. M. Alimi, "Multiple Learned Dictionaries Based Clustered Sparse Coding for the Super-Resolution of Single Text Image," in *2013 12th International Conference on Document Analysis and Recognition*. Washington, DC, USA: IEEE, Aug. 2013, pp. 484–488, http://ieeexplore.ieee.org/document/6628668/.

[48] T. Guha and R. K. Ward, "Learning Sparse Representations for Human Action Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 8, pp. 1576–1588, Aug. 2012.

[49] J. Wang, Y. Yashima, M. Watanabe, and A. Shimizu, "A new sparse representation image coding using multiple bases sets based on image features," in *IWAIT*, Tainan, Taiwan, Jan. 2015.

[50] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004, http://link.springer.com/10.1023/B:VISI.0000029664.99615.94.

[51] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in *Proceedings of the International Conference on Multimedia - MM '10*. New York, New York, USA: ACM Press, 2010, p. 1469, http://dl.acm.org/citation.cfm?doid=1873951.1874249.

[52] J. Macqueen, "Some methods for classification and analysis of multivariate observations," in *5-Th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[53] B. Olshausen and D. Field, "Natural image statistics and efficient coding," *Network: Computation in Neural Systems*, vol. 7, no. 2, pp. 333–339, May 1996, http://www.informaworld.com/openurl?genre=article&doi=10.1088/0954-898X/7/2/014&magic=crossref%7C%7CD404A21C5BB053405B1A640AFFD44AE3.

[54] "HDTV test materials for assessment of picture quality," *The Institute of Image Information and Television Engineering*, Sep. 2009, http://www.ite.or.jp/content/test-materials/.

[55] "Ultra-high definition/wide-color-gamut standard test images," *The Institute of Image Information and Television Engineering*, 2014, http://www.ite.or.jp/content/test-materials/uhdtv/.

[56] ISO/IEC 10918-1, *Information Technology – Digital Compression and Coding of Continuous-Tone Still Images: Requirements and Guidelines*, 1994, https://ci.nii.ac.jp/naid/10029957267.

[57] ISO/IEC 13818-2, *Information Technology – Generic Coding of Moving Pictures and Associated Audio Information*, 2000, https://ci.nii.ac.jp/naid/20000143617.

[58] ISO/IEC 23008-2, *Information Technology – High Efficiency Coding and Media Delivery in Heterogeneous Environments – Part 2: High Efficiency Video Coding*, 2017, http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/96/69668.html.

[59] J. Sole, R. Joshi, N. Nguyen, T. Ji, M. Karczewicz, G. Clare, F. Henry, and A. Duenas, "Transform Coefficient Coding in HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1765–1777, Dec. 2012.

[60] ISO/IEC 14496-10:2014, *Information Technology — Coding of Audio-Visual Objects — Part 10: Advanced Video Coding*, 8th ed., Sep. 2014, https://www.iso.org/standard/66069.html.

[61] N. Pati, A. Pradhan, L. K. Kanoje, and T. K. Das, "An Approach to Image Compression by Using Sparse Approximation Technique," *Procedia Computer Science*, vol. 48, pp. 769–775, Jan. 2015, http://www.sciencedirect.com/science/article/pii/S187705091500722X.

[62] D. Webb, "The statistics of relative abundance and diversity," *Journal of Theoretical Biology*, vol. 43, no. 2, pp. 277–291, Feb. 1974, https://linkinghub.elsevier.com/retrieve/pii/S0022519374800603.

[63] C.-Y. Wang, S.-M. Lee, and L.-W. Chang, "Designing JPEG quantization tables based on human visual system," *Signal Processing: Image Communication*, vol. 16, no. 5, pp. 501–506, Jan. 2001, http://www.sciencedirect.com/science/article/pii/S0923596500000126.

[64] ITU-T Recommendation P.910, "Subjective video quality assessment methods for multimedia applications," https://www.itu.int/rec/T-REC-P.910-200804-I, Apr. 2008.

[65] S. A. Glantz and B. K. Slinker, *Primer of Applied Regression and Analysis of Variance*. New York: McGraw-Hill, Health Professions Division, 1990.

[66] S. M. Stigler, "Francis Galton's Account of the Invention of Correlation," *Statistical Science*, vol. 4, no. 2, pp. 73–79, May 1989, http://projecteuclid.org/euclid.ss/1177012580.