**Ministry of Higher Education and Scientific Research**
**وزارة التعليم العالي و البحث العلمي**

**Badji Mokhtar Annaba University**
**Université Badji Mokhtar – Annaba**
**Faculty of Technology**
**Department of Electronics**

**جامعة باجي مختار عنابة**

**كلية التكنولوجيا**
**قسم الإلكترونيك**

# Thesis

Submitted to obtain the diploma of

## Doctorate Third Cycle

**Field**: Telecommunication

**Specialty**: Signals and telecommunication systems

By
**Mefoued Abdelkader**

Title

**Encodeur de faible consommation énergétique pour la surveillance embarquée.**

Thesis defended on **06/11/2025** in front of the jury composed of:

| N° | Name and Surname | Grade | Establishment | Role |
|---|---|---|---|---|
| 1 | **LAFIFI SADDEK** | Prof. | Badji Mokhtar University-Annaba | President |
| 2 | **KOUADRIA NASREDDINE** | Prof. | Badji Mokhtar University-Annaba | Supervisor |
| 3 | **HARIZE SALIHA** | Prof. | Badji Mokhtar University-Annaba | Co-supervisor |
| 4 | **SERIR AMINA** | Prof. | U.S.T.H.B | Examiner |
| 5 | **BEKHOUCH AMARA** | Prof. | Mohamed Cherif Messaadia University-Souk Ahras | Examiner |
| 6 | **DOGHMANE NOUREDDINE** | Prof. | Badji Mokhtar University-Annaba | Invited |

# Low power encoder for embedded monitoring.


## Abstract

This thesis, titled "Low Power Encoder for Embedded Monitoring," focuses on the development of efficient image compression techniques tailored to the needs of embedded monitoring systems. These systems, often constrained by limited energy and computational resources, require innovative approaches to ensure effective data processing and transmission. The study emphasizes low-complexity and energy-efficient algorithms utilizing the Discrete Cosine Transform (DCT) and Discrete Tchebichef Transform (DTT), which are widely used for data compression in digital media, telecommunications, and storage systems.

A novel 8-point DCT approximation is proposed, achieving up to a 1 dB improvement in image quality compared to existing methods while maintaining a computational structure optimized for energy-sensitive embedded monitoring applications. A pruned version of this DCT further enhances the trade-off between performance and efficiency. Additionally, two new DTT approximations are introduced, reducing computational complexity and improving compression performance. These approximations are validated through FPGA implementations, demonstrating superior hardware performance and energy efficiency, with a quality gain of up to 2 dB compared to state-of-the-art DTT methods.

Furthermore, the thesis presents a novel algorithm for generating customized quantization tables and coefficient orderings specifically designed for the proposed DCT and DTT approximations. This algorithm consistently improves image quality, delivering an average PSNR increase of 0.3 dB, thereby enhancing the overall performance of the encoder.

The contributions of this research provide significant advancements in the design of low-power encoders for embedded monitoring systems. By addressing the challenges of energy efficiency and computational complexity, the proposed methods offer practical solutions for modern embedded systems and other resource-constrained signal processing applications.


**Keywords:** Low power encoder, Embedded monitoring, DCT/DTT approximations, JPEG, Quantization, Low-complexity algorithms, Lossy compression

# Encodeur de faible consommation énergétique pour la surveillance embarquée.

## Résumé

Cette thèse, intitulée "Encodeur de Faible Consommation énergétique pour la Surveillance Embarquée," explore les avancées en compression d'images avec un accent particulier sur des techniques à faible complexité et efficacité énergétique adaptées aux systèmes de surveillance embarqués. L'étude porte sur la Transformée en Cosinus Discrète (DCT) et la Transformée de Tchebichef Discrète (DTT), essentielles pour une représentation efficace des données dans les médias numériques, les télécommunications et les systèmes de stockage. L'objectif est de développer des algorithmes novateurs répondant aux contraintes computationnelles et énergétiques des environnements à ressources limitées, tels que les applications embarquées et en temps réel.

Une nouvelle approximation 8 points de la DCT est proposée, offrant une amélioration de qualité d'image allant jusqu'à 1 dB par rapport aux méthodes existantes tout en conservant une structure computationnelle adaptée aux applications sensibles à l'énergie. Une version élaguée de cette approximation est également introduite pour améliorer davantage le compromis entre efficacité et performance. De plus, deux nouvelles approximations de la DTT sont développées, permettant des réductions significatives de complexité computationnelle et une performance accrue en compression. Ces méthodes sont validées par des implémentations sur FPGA, démontrant des performances matérielles et une efficacité énergétique supérieures, avec un gain de qualité atteignant 2 dB par rapport aux approximations DTT les plus avancées.

La thèse présente également un algorithme innovant pour générer des tables de quantification et des ordonnancements de coefficients personnalisés, optimisés pour des approximations spécifiques de la DCT et de la DTT. Cet algorithme améliore constamment la qualité des images, atteignant une augmentation moyenne du PSNR de 0,3 dB.

Globalement, cette recherche offre des avancées significatives en compression d'images, proposant des solutions à faible complexité et haute efficacité pour les systèmes de surveillance embarqués modernes et d'autres applications de traitement du signal. Ces contributions sont cruciales pour relever les défis d'efficacité énergétique et de contraintes computationnelles dans des environnements limités en ressources.

**Mots-clés :** Approximations DCT/DTT, JPEG, Quantification, Systèmes embarqués, Algorithmes à faible complexité, Compression avec perte

# مشفر منخفض الطاقة للمراقبة المدمجة.

## ملخص

تتناول هذه الأطروحة، بعنوان "مشفّر منخفض الطاقة للمراقبة المدمجة"، التطورات في ضغط الصور مع تركيز خاص على تقنيات ذات تعقيد منخفض وكفاءة طاقة عالية مصممة خصيصًا لأنظمة المراقبة المدمجة. تركز الدراسة على تحويلة جيب التمام المنفصلة (DCT) وتحويلة تشيبيشيف المنفصلة ،(DTT) اللتين تعدان أساسيتين لتمثيل البيانات بشكل فعال في الوسائط الرقمية والاتصالات وأنظمة التخزين. الهدف هو تطوير خوارزميات مبتكرة تعالج القيود الحاسوبية والطاقة في البيئات محدودة الموارد، مثل التطبيقات المدمجة وفي الوقت الحقيقي.

تم اقتراح تقريب جديد لتحويلة DCT ذات 8 نقاط، يوفر تحسيناً في جودة الصورة يصل إلى 1 ديسيبل مقارنة بالطرق الحالية مع الحفاظ على بنية حاسوبية مناسبة للتطبيقات الحساسة للطاقة. كما تم تقديم نسخة مختصرة من هذا التقريب لتعزيز التوازن بين الكفاءة والأداء بشكل أكبر. بالإضافة إلى ذلك، تم تطوير تقريبات جديدة لتحويلة ،DTT مما يحقق تخفيضات كبيرة في التعقيد الحاسوبي وتحسين الأداء في ضغط البيانات. وقد تم التحقق من صحة هذه الأساليب من خلال تنفيذها على ،FPGA مما يظهر أداءً عتاديًا وكفاءة طاقة جيدة، مع تحقيق زيادة في الجودة تصل إلى 2 ديسيبل مقارنة بأحدث تقريبات .DTT

كما تقدم الأطروحة خوارزمية مبتكرة لتوليد جداول تكميم وترتيب معاملات مخصصة، مُحسَّنة لتقريبات معينة لكل من DCT وDTT. تعمل هذه الخوارزمية باستمرار على تحسين جودة الصورة، مما يحقق زيادة متوسطة في PSNR تصل إلى 0.3 ديسيبل.

بشكل عام، تقدم هذه الدراسة تقدمًا كبيرًا في ضغط الصور، حيث توفر حلولًا ذات تعقيد منخفض وكفاءة عالية للأنظمة المدمجة الحديثة للمراقبة وتطبيقات معالجة الإشارة الأخرى. تُعد هذه المساهمات ضرورية لمواجهة تحديات كفاءة الطاقة والقيود الحاسوبية في البيئات ذات الموارد المحدودة.

**الكلمات المفتاحية** : تقريبات DCT/DTT ، JPEG ، التكميم، الأنظمة المدمجة، خوارزمية منخفضة التعقيد، ضغط بفقدان

# Dedication

To my beloved parents, **Ahcen** and **Mounia**, whose sacrifices, unconditional love, and endless support have made all of this possible.

To my brother **Fathi**, whose encouragement has always motivated me to pursue my dreams.

And to my dear sisters, **Radia** and **Bouthaina**, for their unwavering faith in me and their constant love.

To all my friends, whose companionship and support have made this journey so much more meaningful.

This work is dedicated to you all, with immense love and gratitude.

# Acknowledgments

First and foremost, I would like to thank **Allah** for granting me the strength, patience, and wisdom to complete this journey. I dedicate this work to the memory of my late father, **Ahcen**, whose spirit continues to guide me, and to my beloved mother, **Mounia**, for her unconditional love and support throughout my life.

I extend my heartfelt thanks to my brother **Fathi** and my two wonderful sisters for their encouragement and faith in me during this endeavor. A special gratitude goes to my supervisors, **Pr. Nasreddine Kouadria** and **Pr. Saliha Harize**, for their invaluable guidance, expertise, and mentorship throughout the course of this research.

I am deeply grateful to **Pr. Noureddine Doghmane**, whose insights and constructive feedback were crucial in shaping this work. His contributions have had a significant impact on both my personal and professional growth. I also wish to thank **Pr. Moufida Maimour** for her helpful advice and input.

To all my teachers who have shaped my academic path, thank you for your dedication and for instilling in me a lifelong love of learning. I am particularly grateful to **Drici Mohamed**, **Belbel Yacine**, **Matallah Abdallah**, and **Belkhiri Rabah** for their friendship and support during this journey.

I extend my appreciation to my PhD colleagues **Khantouchi Ramzi**, **Senousi Mouhamed**, and **Boutiouta Islam** for their camaraderie, thought-provoking discussions, and constant encouragement. To my colleagues at work, thank you for your unwavering support and understanding during my academic pursuit.

Finally, I would like to express my gratitude to my dissertation committee president, **Pr. Lafifi Saddek**, and the committee members **Pr. Serir Amina**, **Pr. Bekhouch Amara**, and **Pr. Doghmane Noureddine**, for taking the time to evaluate my work and for providing valuable feedback. Their efforts have been instrumental in the completion of this dissertation.

This achievement is the result of the collective efforts of all the wonderful individuals mentioned here, and I will forever remain grateful for their support and encouragement.

# List of Figures

# List of Tables

# List of Abbreviations

**Cg** Coding Gain. 2, 32, 36, 56, 61, 63, 65, 68, 70, 76, 78, 113

**OOC** Optimized Order of Coefficients. 110

**$P_{14}$** Potluri 2014 approximation. ix, 2, 29, 31–33, 36–39, 41–44, 105, 107, 114, 116, 120, 121

**mCg** modified-Coding Gain. xii, 32–34, 36, 45, 113

**AAC** Advanced Audio Coding. 22

**ADC** Analog-to-Digital Converters. 8

**AI** Artificial Intelegent. 11

**APE** absolute percentage error. 36, 43, 114, 116, 119

**ASICs** Application-Specific Integrated Circuits. 8

**BAS** Bouguezel-Ahmad-Swamy. 3, 106, 107, 116, 121

**BWT** Burrows-Wheeler Transform. 19

**CB** Cintra-Bayer approximation. 107, 112–114, 116, 121

**CLB** Configurable Logic Block. 79

**CMCB** Coutinho pruned of MCB. 39, 41, 43–45

**DAC** Digital-to-Analog Converters. 8

**DCT** Discrete Cosine Transform. ix, xi, 1–3, 20–24, 29–31, 33–37, 39–41, 43–45, 55–57, 61, 62, 85, 87, 88, 103–108, 110, 112, 114, 116, 118–122

**DST** Discrete Sine Transform. 103

**DTT** Discrete Tchebichef Transform. xi, 2, 3, 22, 45, 55–68, 70, 76–80, 82–85, 87–90, 93, 94, 103–108, 110, 112, 114, 116, 118, 120–122

**DVFS** dynamic voltage and frequency scaling. 14

**DWT** Discrete Wavelet Transform. 20, 21

**FF** Flip Flops. 79, 95, 96

**FIR** Finite Impulse Response. 35

**FPGA** Field-Programmable Gate Array. 8, 14, 56, 85, 95

**GIF** Graphics Interchange Format. 19, 20

**GPU** Graphics Processing Unit. 14

**HDR** High Dynamic Range. 25

**HVS** human visual system. 106

**IoT** Internet of Things. 7, 11

**ITU-R BT** International Telecommunication Union Radiocommunication Sector Broadcast Television Standards. 23

**JPEG** Joint Photographic Experts Group. x, 1–3, 22–25, 29, 32, 37, 38, 40–43, 55, 57, 72, 80, 85, 87, 97, 98, 104, 105, 107–111, 114–121

**KLT** Karhunen-Loeve transform. 29, 55

**LUT** Look-Up Table. 79, 95, 96

**LZW** Lempel-Ziv-Welch. 19

**MCB** Modified Cintra-Bayer. ix, 1, 29, 30, 32–39, 41–44, 105, 107, 113, 114, 116, 120, 121

**ML** Machine Learning. 11

**MPEG** Moving Picture Experts Group. 22

**MSE** Mean Square Error. xi, 2, 35, 36, 63, 65, 68, 70, 76, 78

**PNG** Portable Network Graphics. 20

**PSNR** Peak Signal-to-Noise Ratio. xi, xii, 36, 41, 43–45, 81–83, 97, 98, 107, 114, 117, 119, 120

**QF** quality factor. 33

**RGB** Red-Green-Blue. ix, 22, 23

**RLE** Run-Length Encoding. 19

**SDCT** Signed Discret Cosine Transform. 65, 69, 70, 78

**SFG** Signal Flow Graph. ix, x, 37, 39, 74, 75

**SSIM** Structural Similarity Index Measure. 41, 43, 81–83, 97, 98, 119

**UQI** Universal Quality Index. 36, 43, 110, 114, 116, 119

**WHT** Walsh-Hadamard Transform. 103

**Wi-Fi** Wireless Fidelity. 9

**YCbCr** Luminance (Y) and Chrominance (Cb and Cr) Color Model. ix, 22, 23

# List of publications


[p1] A. Mefoued, N. Kouadria, S. Harize, and N. Doghmane, "Improving image encoding quality with a low-complexity DCT approximation using 14 additions," *Journal of Real-Time Image Processing*, vol. 20, no. 3, p. 58, 2023.

[p2] A. Mefoued, S. Harize, and N. Kouadria, "Efficient, low complexity 8- point discrete tchebichef transform approximation for signal processing applications," *Journal of the Franklin Institute*, vol. 360, no. 7, pp. 4807–4829, 2023.

[p3] A. Mefoued, N. Kouadria, S. Harize, and N. Doghmane, "Improved discrete tchebichef transform approximations for efficient image compression," *Journal of Real-Time Image Processing*, vol. 21, no. 1, p. 12, 2024.


## Under Review


[ur] A. Mefoued, N. Kouadria, M. Maimour, S. Harize, N. Doghmane, "New Quantization Tables and Coefficient Ordering for Improving Efficiency of Approximate DCT and DTT Transformations", *Journal of Visual Communication and Image Representation*

# Contents

# General Introduction

The rapid advancement of the Internet of Things (IoT) has contributed significantly to its growing adoption across a wide array of industries. Projections suggest that by 2030, over 125 billion devices will be connected globally[1]. A substantial portion of these devices will be part of Wireless Sensor Networks (WSNs), which consist of autonomous sensor nodes that capture, locally process, and transmit data wirelessly. These networks function without relying on any predefined infrastructure, rendering them particularly suitable for diverse embedded monitoring applications, ranging from environmental surveillance to industrial monitoring systems.

This thesis focuses on the deployment of WSNs in the context of **embedded monitoring systems**, wherein multiple sensors, such as acoustic, image, or presence sensors, are utilized in concert to gather and relay data. A prime application of these systems is the real-time monitoring of industrial processes or environmental variables. However, such applications face critical challenges related to the efficient transmission and storage of large volumes of sensor data, especially in resource-constrained environments where power, bandwidth, and storage capacities are limited. Therefore, this thesis proposes low-complexity coding and compression techniques to facilitate efficient data handling in these resource-constrained embedded monitoring systems.

## Problem Statement

Embedded monitoring systems have become essential components across various industries, including manufacturing, agriculture, and environmental monitoring, where continuous observation and data collection are pivotal in maintaining operational efficiency, safety, and sustainability. The utilization of WSNs enables the acquisition of large-scale sensor data autonomously, eliminating the need for human intervention and providing several advantages, such as real-time monitoring, remote access, and cost-effectiveness. However, the resource-constrained nature of WSNs presents several technical challenges, particularly in the following areas:

- **Energy Efficiency:** Since WSNs nodes are often powered by batteries, optimizing energy consumption is paramount to extending the network's operational lifespan.

- **Data Compression and Transmission:** Given the limited bandwidth, efficient compression techniques must be developed to reduce the size of transmitted data without sacrificing the integrity of critical information.

- **Hardware Resources:** Embedded monitoring systems often operate within strict hardware limitations, particularly in terms of memory, processing power, and communication capabilities. The design of compression algorithms must account for these constraints to ensure that the system can perform efficiently within the available hardware resources. Moreover, techniques that can minimize hardware resource utilization, such as low-complexity transformations, are critical for ensuring real-time performance in embedded systems.

This thesis aims to develop innovative compression techniques tailored to address the inherent resource limitations in embedded systems within WSNs. By focusing on compression algorithms designed to reduce both the computational complexity and energy consumption associated with data handling, this work contributes to the advancement of more efficient embedded monitoring systems.

# Thesis Contributions and Objectives

The primary objective of this thesis is to propose and evaluate effective coding and compression techniques for embedded monitoring applications. The specific objectives addressed in this thesis are as follows:

- **Comparative Study of Existing Techniques:** A thorough comparative analysis will be conducted on the state-of-the-art compression and coding techniques employed at the sensor node level. The analysis will focus on identifying methods that minimize computational complexity and energy consumption, while also highlighting the trade-offs associated with various approaches in order to inform the selection of optimal algorithms for embedded monitoring systems.

- **Proposal of a Novel Compression Methodology:** Novel compression methodologies will be introduced to address the challenges of complexity and energy efficiency in resource-constrained environments. These methodologies will aim to substantially reduce computational complexity while maintaining or enhancing image quality in the resulting data streams. Special attention will be given to optimizing both the transformation and quantization stages to achieve high performance within constrained hardware and power

environments.

- **Performance Evaluation:** The proposed compression techniques will undergo rigorous evaluation through a combination of simulations and real-world experimental setups. Key performance indicators, including energy efficiency, data size (bitrate), and image quality (assessed using metrics such as PSNR and Structural Similarity Index Measure (SSIM)), will be examined. The performance results will be critically analyzed to assess the viability of the proposed techniques in real-time embedded monitoring scenarios, with a particular emphasis on reducing energy consumption without compromising data fidelity.

- **Field-Programmable Gate Array (FPGA) Implementation:** The feasibility of the proposed compression methodologies will be validated through their implementation on real-world embedded monitoring systems using FPGA technology. The FPGA platform will serve as an ideal testing environment to demonstrate the practical application of the proposed algorithms in resource-constrained settings. Key metrics, such as power consumption, processing speed, and hardware resource utilization, will be measured to confirm the benefits of the methodology with respect to both energy efficiency and image quality.

## Structure of the Thesis

The remainder of this thesis is organized as follows:

- **Chapter 1** explores the domain of embedded monitoring systems and presents a review of existing methodologies and techniques. A particular focus is placed on image-based embedded monitoring, emphasizing its significance, applications, and the inherent challenges faced in resource-constrained environments such as WSNs and embedded systems.

- **Chapter 2** provides a comprehensive review of the existing methods and technologies for data compression and coding in embedded systems. Special consideration is given to lossy compression techniques, particularly JPEG, in the context of WSNs and IoT applications, where energy consumption constraints necessitate the development of low-complexity algorithms.

- **Chapter 3** presents the proposed low-complexity DCT approximation, detailing the mathematical formulation and algorithmic implementation. This approximation achieves the lowest reported computational complexity, requiring only 14 additions, while outperforming other methods in terms of image quality.

- **Chapter 4** introduces novel low-complexity DTT approximations, demonstrating their efficiency in image compression by striking an optimal balance

between image quality and computational complexity, making them highly suitable for embedded systems with real-time constraints.

- **Chapter 5** describes a novel implementation of the proposed DTT approximation aimed at reducing energy consumption in FPGA-based systems while enhancing image quality. Validation results and performance analyses are provided to substantiate the efficiency and practical utility of the implementation in embedded monitoring applications.

- **Chapter 6** introduces a new algorithm for generating optimized quantization tables and coefficient ordering schemes for each of the proposed approximations, as well as other low-complexity approximations from the literature. This optimization aims to further enhance image quality in resource-limited environments.

- Finally, the **General Conclusion** summarizes the findings of the thesis, evaluates the limitations of the proposed methodologies, and suggests avenues for future research, particularly in the domains of computational complexity reduction and performance optimization for embedded monitoring systems.

Through this research, the efficiency of embedded monitoring systems will be significantly enhanced by the proposed compression techniques, which aim to reduce data transmission requirements, increase energy efficiency, and improve data accuracy.

# Part I

# Background and Motivation

# Chapter 1

# Embedded monitoring

## Introduction

In recent years, the rapid advancement of technology has led to the proliferation of embedded systems across various domains, including healthcare [2], industrial automation [3], and environmental monitoring [4]. Embedded monitoring, defined as the integration of monitoring capabilities within embedded devices, plays a pivotal role in enhancing system performance, safety, and reliability. This chapter explores the fundamental concepts, methodologies, and applications of embedded monitoring, highlighting its significance in contemporary technological landscapes.

Embedded monitoring systems are designed to operate autonomously within larger systems, utilizing microcontrollers and sensors to continuously collect and analyze data. These systems enable real-time monitoring of critical parameters such as temperature, pressure, and humidity, which are essential for maintaining optimal operational conditions. For instance, in healthcare, embedded monitoring devices are employed to track vital signs, providing healthcare professionals with timely information that can lead to improved patient outcomes [2].

The importance of real-time data acquisition cannot be overstated. By facilitating immediate responses to changing conditions, embedded monitoring systems significantly reduce the risks associated with equipment failures and environmental hazards. In industrial settings, predictive maintenance strategies enabled by embedded monitoring can lead to substantial cost savings by minimizing unplanned downtime and extending the lifespan of machinery [5]. Furthermore, the integration of IoT technologies with embedded monitoring systems has transformed traditional monitoring approaches, allowing for remote data access and advanced analytics [6].

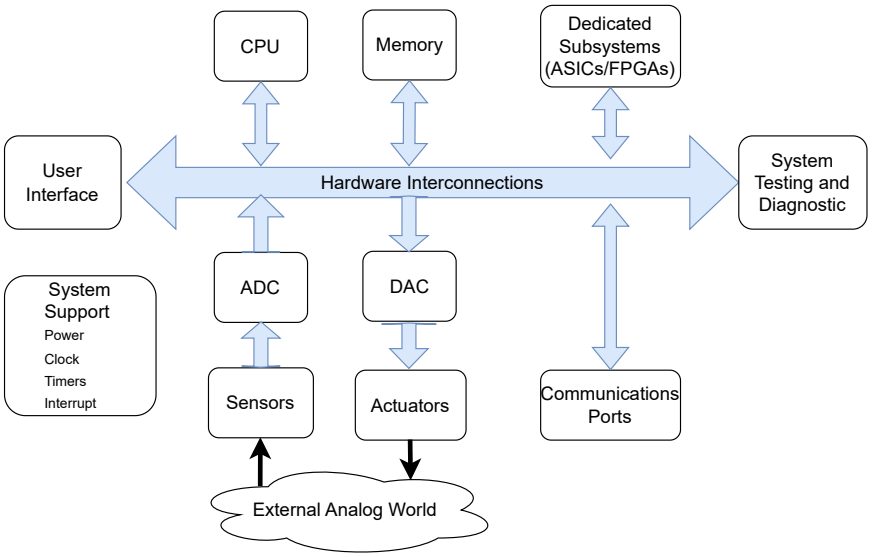



Figure 1.1: Hardware elements in an embedded system

This chapter is structured as follows: First, we will delve into the technical architecture of embedded monitoring systems, including hardware components and software frameworks. Next, we will examine various applications across different sectors, emphasizing the benefits and challenges associated with their implementation. Finally, we will discuss future trends and research directions in embedded monitoring, particularly in the context of IoT and smart technologies. By understanding the principles and applications of embedded monitoring, researchers and practitioners can better leverage these systems to enhance operational efficiency, safety, and sustainability across diverse fields.

## 1.1 Technical Architecture of Embedded Monitoring Systems

Embedded monitoring systems are designed to continuously track and report various parameters in real-time environments, often within constrained resources. These systems are typically powered by batteries, making energy efficiency a critical consideration in their design.

The technical architecture of such systems generally consists of several key components presented in Fig 1.1:

- **Microcontroller/Processor (CPU) Fig.1.2:** At the core of the

system lies a low-power microcontroller or processor that processes the sensor data. The choice of the microcontroller is crucial, as it needs to balance processing power with energy consumption. Modern architectures often utilize sleep modes or dynamic frequency scaling to reduce power usage during idle periods.

- **Dedicated Subsystems (ASICs/FPGAs):** For specialized tasks requiring high efficiency and performance, Application-Specific Integrated Circuits (ASICs) or FPGA as presented in Fig. 1.2 are often integrated into the system. These dedicated subsystems handle specific functions such as data processing or compression, offering greater power efficiency and performance optimization compared to general-purpose processors.

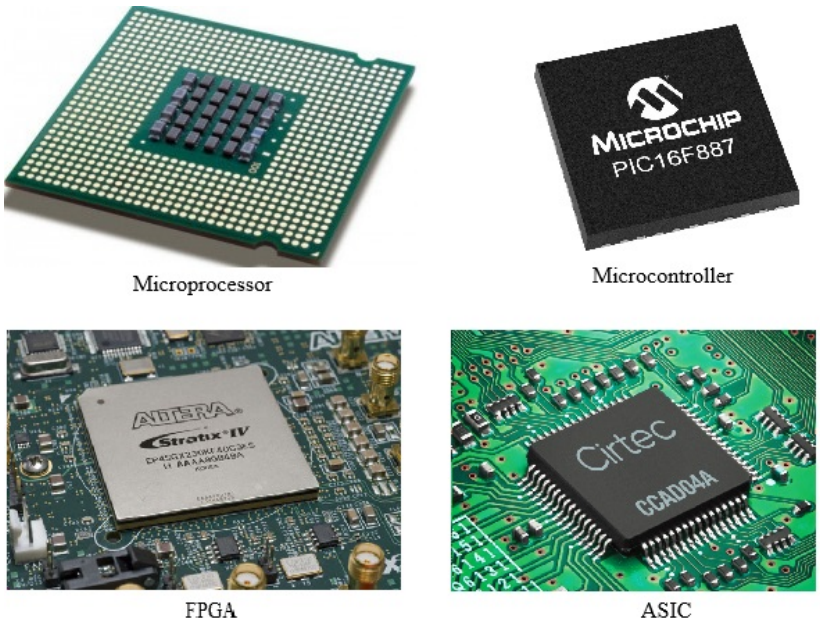



Figure 1.2: Examples of hardware platforms used in embedded systems: Microprocessor, Microcontroller, FPGA, and ASIC.

- **ADC/DAC:** ADC and DAC as shown in Fig. 1.3 play a crucial role in systems where analog signals from sensors must be converted into digital form for processing, or where digital signals must be converted back to analog for interfacing with actuators or other analog devices. These converters must be energy-efficient and designed to match the system's performance requirements.

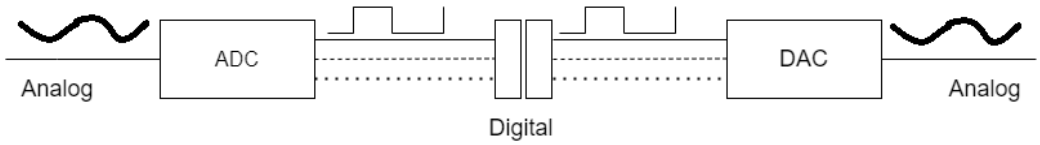



Figure 1.3: A basic representation of ADC and DAC conversion processes.

- **Sensor Interface:** The system is equipped with various sensors that capture environmental or operational data, such as temperature, humidity, vibration, or cameras as presented in Fig. 1.4. These sensors are directly interfaced with the microcontroller or processor for data acquisition.

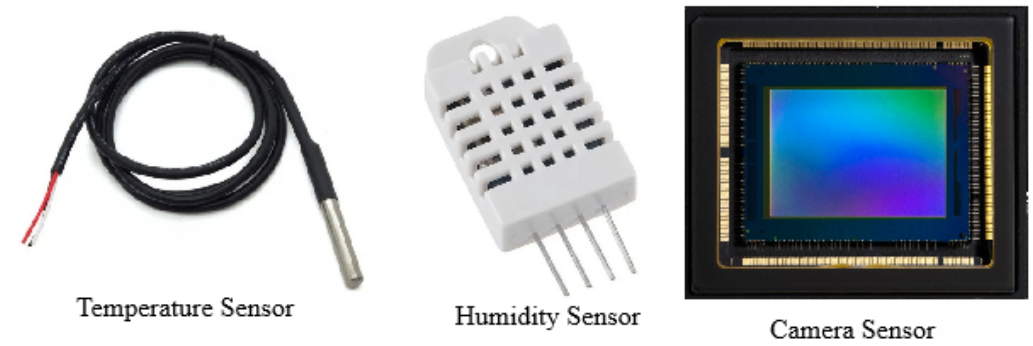



Figure 1.4: Some examples of Sensors used in embedded monitoring systems.

- **Actuators:** Many embedded systems control actuators such as motors, valves, or other mechanical devices as shown in Fig. 1.5. These actuators convert the digital output of the system into physical actions. Proper control and power management are necessary to ensure that actuators operate efficiently while conserving energy, especially in battery-powered systems.
- **Wireless Communication Module:** Many embedded monitoring systems include a wireless communication module (e.g., Wireless Fidelity (Wi-Fi), Bluetooth, Zigbee Fig. 1.6) to transmit data to a central server or cloud platform. This module is one of the most energy-intensive components, so strategies like duty cycling, where the module is powered off when not in use, are implemented to conserve battery life.
- **Power Management:** Effective power management is essential to maximize battery life. This typically involves the use of voltage regu-

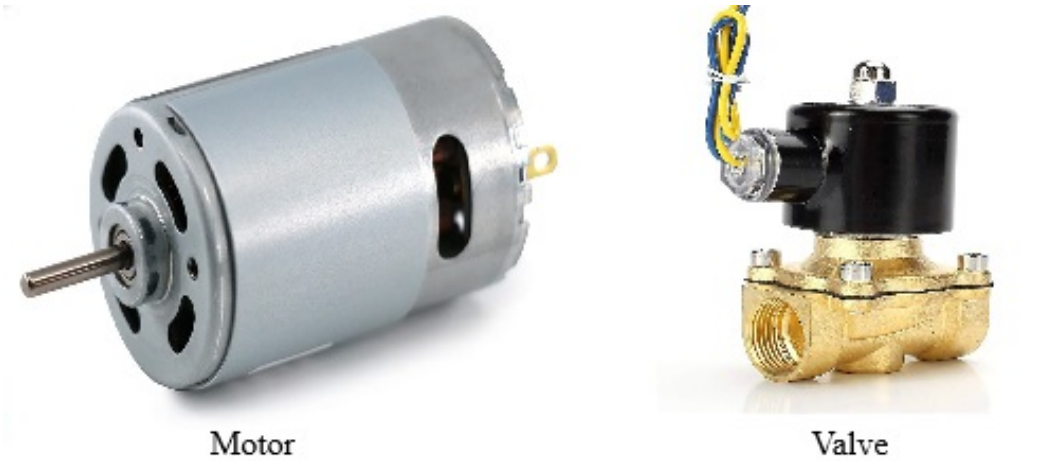



Figure 1.5: Examples of Actuator used in embedded monitoring systems.

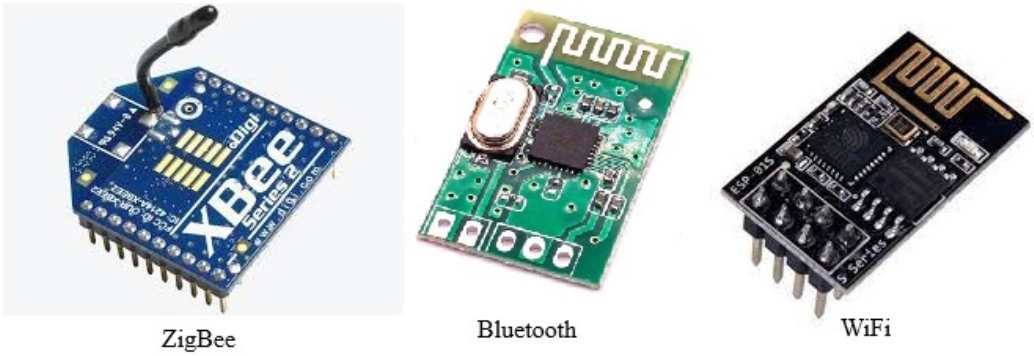



Figure 1.6: Wireless communication modules used in embedded monitoring systems.

lators, power gating techniques, and energy-efficient components. Additionally, the system's software must be optimized to minimize power consumption by ensuring that high-energy tasks are executed efficiently and that low-power modes are utilized whenever possible.

- **Energy Storage (Battery):** The choice of battery technology (e.g., Li-ion, NiMH Fig. 1.7) depends on the power requirements and operational lifetime of the system. The architecture must include battery monitoring circuits to manage charge levels and extend battery life. In some cases, energy harvesting techniques may be employed to supplement battery power.

- **Data Storage:** Non-volatile memory, such as flash storage, is used to store the collected data locally. This is particularly useful in scenarios


Figure 1.7: Widely used batteries in embedded monitoring systems.

where continuous connectivity is not available, allowing the system to buffer data and transmit it later.

- **System Testing and Diagnostics:** An important aspect of the architecture is the inclusion of diagnostics and testing capabilities. These subsystems help detect faults, ensure correct system behavior, and provide feedback for maintenance purposes. In embedded systems, this can involve self-checks and remote monitoring to identify any issues without the need for manual intervention.
- **User Interface (Optional):** Some systems may include a minimal user interface, such as LEDs or a small display, to provide immediate feedback or system status. This interface is typically designed to consume minimal power, activating only when necessary.

## 1.2 History of Embedded Systems

Embedded systems have played a significant role in technological advancements over the past several decades. Their development closely follows the evolution of microprocessors and electronic components. Below is a brief overview of the key stages in the history of embedded systems:

### The Early Years (1950s-1960s):

- **Use of Discrete Components:** Early embedded systems were constructed using individual components like transistors and diodes. These systems were often bulky, consumed a lot of power, and were quite limited in their capabilities [7].

- **Specialized Systems:** Examples from this period include military guidance systems and industrial control computers used in specific applications [7].

### The Emergence of Microprocessors (1970s):

- **Intel 4004 Introduction:** The 1971 launch of the Intel 4004 was a major turning point. As the first commercially available microprocessor, it provided a more efficient and compact solution for embedded system designs [8].
- **Applications in Consumer Electronics:** Soon after, microprocessors found widespread use in consumer products such as calculators and early video game systems [9].

### Advances in Microcontroller Technology (1980s-1990s):

- **The Rise of Microcontrollers:** The introduction of microcontrollers, which integrated the microprocessor, memory, and input/output peripherals onto a single chip, simplified the development of embedded systems [10].
- **Widespread Adoption:** Microcontrollers became common in automotive systems, home appliances, and other everyday consumer products [10].

### The Modern Era (2000s-Present):

- **The IoT:** With the advent of IoT, embedded systems have become central to devices ranging from smart home appliances to industrial sensors [11].
- **Integration of Artificial Intelegent (AI) and Machine Learning (ML):** Embedded systems are increasingly incorporating capabilities such as artificial intelligence and machine learning, enabling more sophisticated and autonomous applications [12].

## 1.3 Image-based Monitoring Systems

Image-based monitoring has emerged as a transformative approach in various fields, leveraging visual data to enhance decision-making, improve safety,

and optimize performance. This method utilizes advanced image processing techniques to analyze visual information captured through cameras and sensors, enabling real-time monitoring of conditions in diverse applications such as structural health, environmental assessment, and public safety. As embedded monitoring systems continue to advance, the integration of image-based monitoring has become increasingly crucial in enhancing operational efficiency, safety, and responsiveness.

### 1.3.1 Importance of Image-Based Monitoring in Embedded Systems

- **Structural Health Monitoring:** Image-based monitoring plays a critical role in assessing the integrity of civil infrastructures. By employing non-destructive evaluation techniques, it allows for continuous monitoring of structures such as bridges and buildings without compromising their integrity. This capability is essential for early detection of potential failures, thereby extending the service life of these assets and ensuring public safety [13].

- **Environmental Monitoring:** In environmental applications, image-based monitoring facilitates the assessment of natural resources and ecosystems. For instance, it can be used to monitor vegetation health, track wildlife populations, and assess changes in land use. The ability to analyze large volumes of visual data helps in understanding ecological dynamics and informing conservation efforts [14].

- **Construction Safety:** The construction industry benefits significantly from image-based monitoring through enhanced safety protocols. Techniques such as computer vision can identify hazardous conditions and monitor worker safety in real-time, reducing the incidence of accidents on site. This proactive approach to safety management is crucial in high-risk environments [15].

- **Healthcare Applications:** In healthcare, image-based monitoring systems are invaluable for patient care, enabling continuous observation of vital signs and other health indicators. These systems can alert medical personnel to critical changes, facilitating timely interventions and improving patient outcomes [16].

- **Social Media and Brand Management:** The rise of visual content on social media platforms necessitates effective image-based monitoring for brands. By analyzing user-generated images, companies can gain

insights into customer sentiment and product performance, allowing them to address issues proactively and enhance customer engagement [17].

## 1.4 Challenges of Image-Based Embedded Systems

Image-based embedded systems have become increasingly prevalent in applications such as surveillance, autonomous vehicles, medical imaging, and industrial automation [18, 19]. These systems process and analyze visual data in real-time, often under strict constraints regarding size, power, and performance. Despite their advantages, image-based embedded systems face significant challenges, particularly in terms of energy consumption and storage requirements.

### 1.4.1 Energy Consumption

Processing high-resolution images and videos demands substantial computational power, which directly impacts energy consumption. In battery-powered embedded systems, this becomes a critical issue as the need to balance performance with energy efficiency is paramount. Key factors contributing to high energy consumption include:

- **Complex Image Processing Algorithms:** Tasks such as object detection, image recognition, and feature extraction require intensive computations, often involving multiple stages of processing that consume significant energy [**Horowitz20141**].
- **Continuous Operation:** Many image-based systems, especially those used in surveillance or autonomous navigation, operate continuously. This constant processing leads to sustained energy usage, draining battery resources quickly.
- **High Data Throughput:** The processing of large image files or video streams requires frequent access to memory and data transfer operations, both of which are energy-intensive. The need to maintain high data rates further exacerbates the energy demands [20].

To address these issues, designers often employ strategies such as optimizing algorithms for lower complexity, using specialized hardware accelerators (e.g., Graphics Processing Unit (GPU)s or FPGAs), and implementing power

management techniques like dynamic voltage and frequency scaling (DVFS) or duty cycling. However, these solutions introduce their own trade-offs, such as increased hardware complexity or reduced processing speed.

### 1.4.2 Storage Constraints

The storage of images and video data presents another major challenge in image-based embedded systems. High-resolution images and videos generate vast amounts of data, which must be stored efficiently without compromising the system's performance or longevity.

- **Large Data Volumes:** Images and video frames, particularly in high resolution, require significant storage capacity. For instance, a single 1080p image can consume several megabytes, and when dealing with video streams at high frame rates, the data requirements quickly multiply. Embedded systems, which often have limited storage resources, must manage these large volumes of data effectively.
- **Data Retention and Reliability:** The need for reliable data storage is critical in applications like medical imaging or industrial monitoring, where loss of data could have serious consequences. Embedded systems must ensure that storage media can retain data over long periods without degradation, all while operating within tight energy budgets.
- **Compression and Data Management:** To mitigate storage constraints, image-based embedded systems frequently employ compression techniques to reduce the size of stored data. However, compression algorithms introduce computational overhead, which can impact both energy consumption and processing speed. Additionally, managing compressed data requires careful consideration to avoid excessive latency during data retrieval or decompression.
- **Wear and Longevity of Storage Media:** Flash memory, commonly used in embedded systems, has limited write cycles. The continuous writing and rewriting of large data sets, such as video streams, can lead to faster wear of the storage medium, potentially reducing the system's operational life [21].

To address these storage challenges, designers must strike a balance between data compression, storage capacity, and access speed. Additionally, advanced file systems and wear-leveling techniques can help extend the life of storage media by distributing write operations more evenly across the memory.

## 1.5 Conclusion

The challenges of energy consumption and storage in image-based embedded systems are interrelated and require careful consideration during system design. Solutions must focus on optimizing both the hardware and software to achieve the desired performance while maintaining energy efficiency and managing data storage effectively. As image-based applications continue to evolve, overcoming these challenges will be critical to enabling the next generation of embedded systems [18].

# Chapter 2

# Image and Video compression

## Introduction

Image compression is a fundamental technique in the field of digital image processing, aimed at reducing the amount of data required to represent an image while preserving its essential visual information. As technology advances and the demand for high-quality multimedia content grows, the need for efficient image compression methods becomes increasingly crucial. This chapter explores the significance of image compression, its diverse applications, various compression techniques, and the challenges faced in achieving optimal compression.

The explosive growth of digital imagery in various domains, such as photography, medical imaging, satellite imagery, and multimedia content delivery, has led to a massive accumulation of data. Storing and transmitting such voluminous data demand substantial resources, including storage space and network bandwidth. Image compression addresses this challenge by enabling the representation of images in a more compact form, resulting in reduced storage requirements and faster transmission speeds. Efficient image compression techniques not only save storage space and transmission time but also contribute to cost savings and improved user experiences.

Image compression plays a pivotal role in a wide range of real-world applications. In fields such as remote sensing, surveillance, and monitoring, image compression facilitates the efficient storage and transmission of images captured from remote locations. For instance, monitoring ecosystems and wildlife habitats often involve capturing images from various sensors and cameras placed in remote areas. Efficient image compression ensures that the transmitted data is manageable and can be analyzed effectively for monitoring purposes. This is particularly relevant in environmental studies,

where monitoring the behavior of species and changes in ecosystems requires continuous image acquisition and processing.

Image compression techniques can be broadly classified into two categories: lossless and lossy compression. Lossless compression methods preserve all the original image data during compression and decompression, making them suitable for applications where data integrity is critical, such as medical imaging and archiving. On the other hand, lossy compression methods achieve higher compression ratios by discarding certain non-essential image details. These techniques are often used in applications like multimedia streaming, where some degree of quality loss is acceptable.

While image compression offers significant benefits, it also presents several challenges. One major challenge is finding the right balance between compression ratio and image quality. Achieving higher compression ratios without compromising perceptual image quality requires sophisticated compression algorithms. Additionally, the choice of compression technique must be tailored to the specific application's requirements. Addressing these challenges necessitates ongoing research and development to create more efficient and effective image compression methods.

## 2.1 Importance of image compression

Compressing image and video serves several important purposes:

- Preservation of Storage Space: Image compression significantly reduces the size of image files, conserving storage space on devices and servers.
- Faster Transmission: Compressed images can be transmitted more quickly over networks, making them ideal for web content and streaming.
- Bandwidth Efficiency: Reduced image size leads to lower bandwidth consumption, which is essential for efficient data transfer.
- Enhanced User Experience: Faster loading times for web pages and reduced buffering for multimedia content result in a better user experience.
- Cost Savings: Image compression reduces storage and bandwidth costs, making it economically advantageous for businesses and websites.
- Mobile-Friendly: Compressed images are essential for mobile devices with limited storage and slower internet connections.

- Eco-Friendly: Reduced data transfer and storage requirements contribute to lower energy consumption and a smaller carbon footprint.

In summary, image compression is a valuable technique for optimizing data management, reducing costs, and enhancing overall system performance. It is widely used in various applications, from everyday file compression to more specialized fields like image and video compression. Image compression techniques can be broadly categorized into two main types: lossless and lossy compression.

## 2.2 Lossless Compression

Lossless compression is a data compression technique that preserves all the original information in an image during the compression and decompression process. This ensures that the reconstructed image is an exact replica of the original, making lossless compression suitable for applications where data integrity and fidelity are paramount.

### 2.2.1 Techniques Used in Lossless Compression

Lossless compression employs various techniques to reduce redundancy and minimize data size while enabling precise image reconstruction:

1. **Run-Length Encoding (RLE):** This technique replaces sequences of repeated data with a single data value and a count of the repetition. It is effective for compressing simple repeating patterns or characters.

2. **Huffman Coding:** Huffman coding assigns shorter codes to more frequently occurring symbols and longer codes to less frequent symbols. This technique is widely used in file compression and is based on the frequency distribution of symbols.

3. **Arithmetic Coding:** Similar to Huffman coding, arithmetic coding assigns codes to sequences of symbols based on their probabilities. It often achieves better compression ratios than Huffman coding.

4. **Dictionary-Based Compression:** Techniques like Lempel-Ziv-Welch (LZW) and Burrows-Wheeler Transform (BWT) create a dictionary of frequently occurring patterns or substrings and encode them with shorter codes. LZW is used in Graphics Interchange Format (GIF) and BWT is used in the Unix compression tool "bzip2."

5. **Delta Coding:** Delta coding encodes the difference between consecutive data values, taking advantage of the correlation between nearby values. It is effective for compressing data with gradual changes.

6. **Entropy Coding:** Entropy coding methods, such as Shannon-Fano coding or Arithmetic Coding, exploit the statistical properties of the data to achieve efficient compression. These methods use variable-length codes to represent data with higher probability values using shorter codes.

7. **Predictive Coding:** Predictive coding predicts the value of a data point based on its neighboring values and encodes the prediction error. It is commonly used for compressing audio and image data.

8. **Transform Coding:** Techniques like DCT and Discrete Wavelet Transform (DWT) convert the data into a domain where most of the energy is concentrated, allowing for efficient compression of the transformed coefficients.

9. **Lossless Image Formats:** Specialized formats like Portable Network Graphics (PNG) and GIF use a combination of techniques such as adaptive filtering, entropy coding, and color palette indexing to achieve lossless image compression.

## 2.3 Lossy Compression

Lossy compression techniques are used to reduce the size of data by sacrificing some amount of information or quality. While lossy compression can achieve higher compression ratios compared to lossless compression, it introduces some degree of degradation in the data. Some common techniques used in lossy compression include:

1. **Quantization:** Quantization reduces the precision of data values by mapping them to a smaller set of values. This technique is commonly used in image and video compression, where color and intensity values are quantized to reduce the number of bits required.

2. **Subsampling:** Subsampling is a technique used in image and video processing where the resolution of certain image components, typically color information (chrominance), is reduced. This process retains full detail in the luminance (brightness) while compressing the chrominance, allowing for reduced data size and storage requirements without

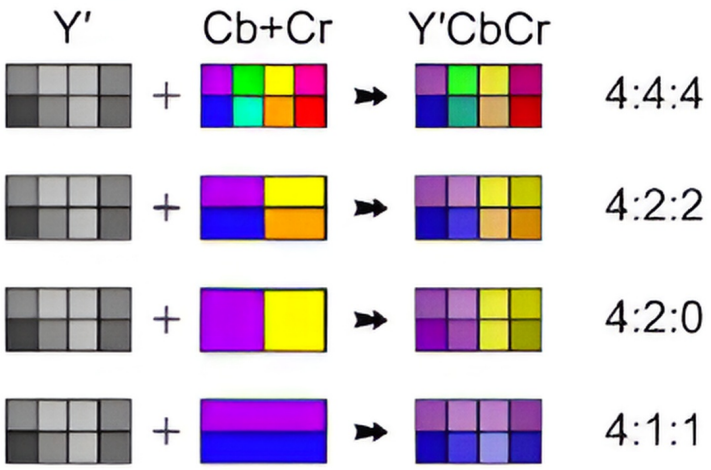



Figure 2.1: Widely used chroma subsampling formats

significantly affecting visual quality. Common subsampling ratios are presented in Fig. 2.1 include 4:2:2, 4:2:0 and 4:1:1 in video compression, where color information is reduced in favor of luminance data.

- 4:4:4 - No subsampling; each channel (Y, Cb, Cr) has the same resolution.
- 4:2:2 - Cb and Cr are sampled at half the horizontal resolution of Y, but the same vertical resolution.
- 4:2:0 - Cb and Cr are sampled at half the horizontal and vertical resolution of Y.
- 4:1:1 - Cb and Cr are sampled at one-quarter of the horizontal resolution of Y, but maintain the same vertical resolution. This means that for every 4 pixels in the horizontal direction, there is only 1 Cb and 1 Cr sample, while Y maintains full resolution.

3. **Psycho-visual Techniques:** These techniques take advantage of limitations in human perception to selectively remove details that are less noticeable to the human eye. This includes techniques like chroma subsampling, masking, and perceptual quantization.

4. **Transform Coding:** Similar to lossless compression, lossy compression techniques also utilize transform coding methods like DCT and DWT. In lossy compression, these transformed coefficients undergo quantization to achieve compression, albeit with a degree of data loss.

This transform step remains lossless when the transform kernel is orthogonal. However, for quasi-orthogonal transform kernels, such as certain DCT approximations and all DTT approximations, this step becomes inherently lossy.

5. **JPEG:** JPEG is a widely used lossy compression standard for images. It uses DCT and quantization to achieve high compression ratios while maintaining reasonable image quality.

6. **Moving Picture Experts Group (MPEG):** MPEG is a family of standards for video compression. Techniques like motion compensation, inter-frame prediction, and quantization are used to achieve efficient compression for video sequences.

7. **Advanced Audio Coding (AAC):** AAC is a lossy audio compression format that uses techniques like psychoacoustic modeling, transform coding, and quantization to achieve high-quality audio compression at lower bit rates.

## 2.4 JPEG Compression

JPEG is a widely used lossy image compression standard that is suitable for photographs and natural images. It employs several techniques to achieve high compression ratios while maintaining reasonable image quality.

### 2.4.1 Algorithm Overview

The JPEG compression algorithm involves the following key steps that are depicted in Fig. 2.2:

1. **Color Space Conversion:** The image is usually converted from the RGB color space (Fig. 2.3) to the YCbCr color space(Fig. 2.4). This separates the luminance (brightness) information from the chrominance (color) information. The formula to convert RGB color values to YCbCr color values is as follows:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}, \tag{2.1}$$

with:

- $Y$ represents the luma (brightness) component.

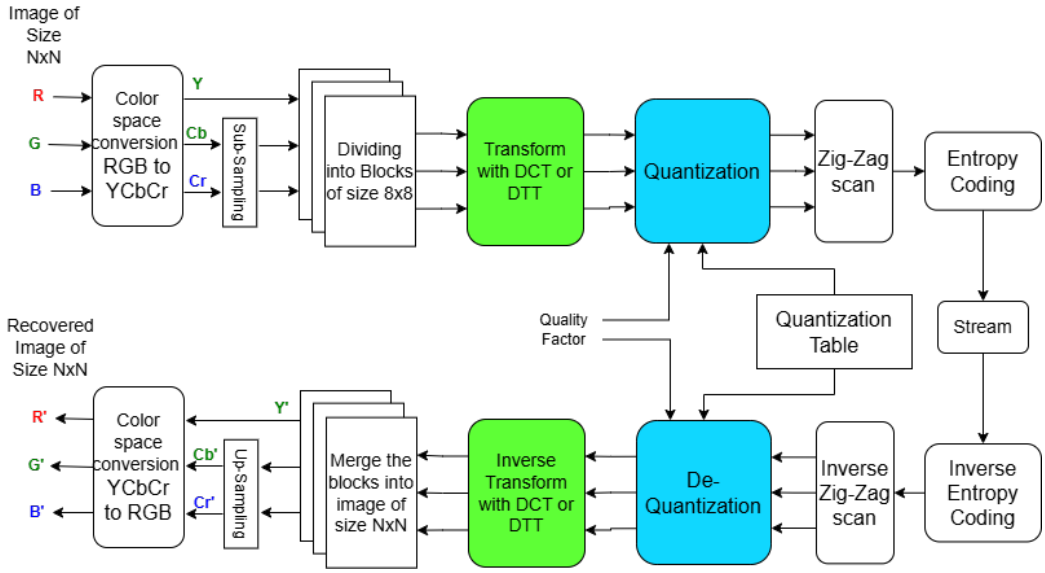



Figure 2.2: Architecture for a JPEG compression system, on top encoding, on bottom decoding

- $Cb$ represents the blue-difference chroma component.
- $Cr$ represents the red-difference chroma component.
- $R$,$G$, and $B$ are the original RGB color values, typically in the range of 0 to 255.

These formulas are used to perform the color space conversion from RGB to YCbCr. Note that these equations are specific to the International Telecommunication Union Radiocommunication Sector Broadcast Television Standards (ITU-R BT).601 standard, which is commonly used for digital video. There are variations and other standards (e.g., ITU-R BT.709) that might use slightly different coefficients.

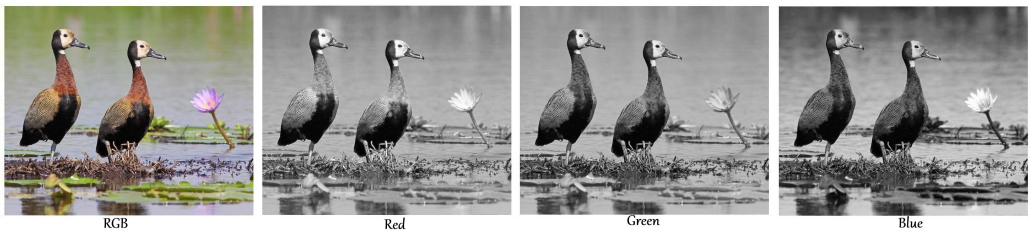



Figure 2.3: Color image with its RGB elements


Figure 2.4: Color image with its YCbCr elements

2. **Dividing the Image into Blocks:** The image is divided into small blocks, typically 8x8 pixels each.

3. **DCT:** A DCT is applied to each block. This transforms the pixel values from the spatial domain to the frequency domain, allowing for efficient compression of the image.

   The DCT is a widely used tool in signal processing for image and video compression. It enables the representation of image data in a more compact format, making it more efficient for storage and transmission. The conventional DCT algorithm involves multiplying the image data with a cosine matrix defined in (2.2). However, this multiplication process can be computationally expensive, making it less suitable for use in real-time applications or systems with limited resources. As a result, researchers have proposed several approximation methods for the DCT that can perform the transformation faster and with fewer computations.

   The following matrix represents the kernel of the DCT transformation:

$$C = \begin{pmatrix} \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 \\ \gamma_0 & \gamma_2 & \gamma_4 & \gamma_6 & -\gamma_6 & -\gamma_4 & -\gamma_2 & -\gamma_0 \\ \gamma_1 & \gamma_5 & -\gamma_5 & -\gamma_1 & -\gamma_1 & -\gamma_5 & \gamma_5 & \gamma_1 \\ \gamma_2 & -\gamma_6 & -\gamma_0 & -\gamma_4 & \gamma_4 & \gamma_0 & \gamma_6 & -\gamma_2 \\ \gamma_3 & -\gamma_3 & -\gamma_3 & \gamma_3 & \gamma_3 & -\gamma_3 & -\gamma_3 & \gamma_3 \\ \gamma_4 & -\gamma_0 & \gamma_6 & \gamma_2 & -\gamma_2 & -\gamma_6 & \gamma_0 & -\gamma_4 \\ \gamma_5 & -\gamma_1 & \gamma_1 & -\gamma_5 & -\gamma_5 & \gamma_1 & -\gamma_1 & \gamma_5 \\ \gamma_6 & -\gamma_4 & \gamma_2 & -\gamma_0 & \gamma_0 & -\gamma_2 & \gamma_4 & -\gamma_6 \end{pmatrix}, \tag{2.2}$$

   where $\gamma_i = \frac{1}{2} \cdot cos(\frac{2\pi(i+1)}{32})$, $i = 0, 1, 2, 3, 4, 5, 6$. The 2D transform domain named $Y$ of an input block of size $8 \times 8$ can be calculated using the following equation:

$$Y = C \cdot X \cdot C^T, \tag{2.3}$$

where $X$ is the input block, $C$ is the DCT kernel defined in (2.2). The original input block of size $8 \times 8$ can be recovered from the DCT transform domain $Y$ by using the following equation:

$$X = C^T \cdot Y \cdot C \tag{2.4}$$

4. **Quantization:** The DCT coefficients are quantized to reduce the number of bits required to represent them. This is a lossy step, as it involves rounding off the coefficients. JPEG compression uses quantization tables to control the amount of compression applied to different frequency components. Different quantization tables can be used for luminance (2.5) and chrominance (2.6) data, allowing for different compression levels for different color channels.

$$Q_L = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 65 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \tag{2.5}$$

$$Q_C = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix} \tag{2.6}$$

5. **Entropy Coding:** The quantized coefficients are encoded using variable-length entropy coding techniques, such as Huffman coding or arithmetic coding.

### 2.4.2 Quality Factors (QF)

JPEG compression offers users the flexibility to adjust the compression quality by selecting different quantization tables or quality settings, also referred to as the QF. Higher quality factors result in better image quality but increase file sizes, while lower quality factors reduce file sizes at the cost of some loss in image detail.

$$Q = \begin{cases} Q_0 & \text{if } QF = 50 \\ round((Q_0 \cdot SF + 50) \div 100) & \text{otherwise ,} \end{cases} \tag{2.7}$$

here, $Q_0$ is replaced by $Q_L$ for luminance components or $Q_C$ for chrominance components, with $0 < QF \leq 100$. The scaling factor $SF$ is defined as follows:

$$SF = \begin{cases} 5000 \div QF & \text{if } QF < 50 \\ 200 - 2 \times QF & \text{if } QF > 50. \end{cases}$$

The quantization process is then performed as follows:

$$Y_Q = round(Y \oslash Q), \tag{2.8}$$

where $Y$ is the transformed domain of a single $8 \times 8$ block as defined in (2.3), and $Q$ is the quantization table based on the selected $QF$, as defined in (2.7).

Since quantization is a lossy process, the original block $\hat{Y}$ can only be approximated through de-quantization, which is defined as follows:

$$\hat{Y} = Y_Q \otimes Q, \tag{2.9}$$

### 2.4.3 JPEG Applications

JPEG compression is one of the most widely adopted image compression standards, thanks to its ability to significantly reduce file sizes while maintaining acceptable image quality. Its applications span a wide range of fields, including but not limited to:

- **Digital Photography:** JPEG is the preferred format for digital cameras and smartphones due to its efficient compression, enabling users to store large numbers of photos without consuming excessive storage space. The balance between file size and image quality makes it ideal for consumer use.
- **Web and Social Media:** JPEG is extensively used in web applications and social media platforms [17], where images need to be transmitted quickly over networks with varying bandwidths. It enables fast loading times for web pages and supports the sharing of photos across social networks without significantly compromising quality.
- **Medical Imaging:** While lossless compression is often used in sensitive applications like medical imaging [2, 22, 23], JPEG is still applicable in areas where slight loss in quality is acceptable, such as image sharing and archiving, where file size is a concern.

- **Digital Art and Graphics:** For web graphics, JPEG allows artists and designers to save artwork with minimal data, making it ideal for portfolios, advertisements, and banners where maintaining a balance between image quality and file size is necessary.
- **Image Archiving and Databases:** JPEG is used for archiving images, allowing large image collections to be stored without overwhelming storage capacities. This is particularly important in sectors like journalism, research, and public archives where file size matters.
- **Satellite and Surveillance Systems:** In remote sensing [24] and surveillance [25, 26], JPEG is used to compress images captured by satellites or security cameras. The ability to store large quantities of images with reduced storage needs is essential in such applications, although some compromise in image quality may be acceptable.
- **Embedded Systems:** JPEG compression is often used in embedded systems [27], such as smart cameras, drones, and portable devices, where storage space and processing power are limited. The ability to reduce image sizes while maintaining adequate quality is critical for resource-constrained environments.
- **IoT:** JPEG is widely used in IoT applications [28], particularly in devices that capture and transmit images, such as smart home security systems and connected sensors. Efficient compression ensures that images can be transmitted over networks with limited bandwidth and processed by cloud services with minimal latency.

Overall, JPEG compression remains a vital technology for reducing image sizes across a variety of fields, enabling easier storage, transmission, and sharing of images, while maintaining sufficient visual fidelity for the majority of practical purposes.

### 2.4.4 JPEG Extensions

While the basic JPEG standard remains widely used, several extensions and variants have been developed to address specific needs and improve upon the original standard. These extensions offer enhanced features, such as better compression efficiency, support for High Dynamic Range (HDR), and lossless compression. Some of the most notable JPEG extensions are:

- **JPEG 2000:** Introduced in the early 2000s, JPEG 2000 is an advanced version of the original JPEG format, providing superior image quality

at lower bitrates. It uses wavelet-based compression (DWT) instead of the DCT, allowing for better compression ratios, lossless compression, and progressive decoding. JPEG 2000 is used in areas requiring high-quality images, such as digital cinema, medical imaging, and satellite imagery.

- **JPEG-LS:(1999)** JPEG-LS is designed specifically for lossless and near-lossless compression, achieving higher efficiency for applications where quality loss is unacceptable. It is commonly used in medical imaging, satellite imagery, and other fields where precise reproduction of images is crucial.

- **JPEG XR(2009):** JPEG XR (formerly known as high-definition (HD) Photo) is a more recent extension that aimed at providing better support for higher resolution and higher bit-depth images, such as those found in HDR content. It offers both lossy and lossless compression options and is optimized for efficient memory usage, making it suitable for applications in photography, image editing, and HD content distribution.

- **JPEG XT(2015):** JPEG XT is an extension of the original JPEG standard that brings new capabilities such as HDR image encoding. It maintains backward compatibility with traditional JPEG decoders, ensuring that legacy systems can still display JPEG XT images, albeit without the enhanced HDR features. JPEG XT allows for an improved dynamic range and color depth, making it suitable for applications like HDR photography and digital archiving.

- **JPEG-HDR(2015):** This extension supports HDR imaging, allowing for better representation of images with a wide range of brightness levels. JPEG-HDR is primarily targeted at photography and imaging fields where color accuracy and dynamic range are critical.

- **JPEG XS(2019):** JPEG XS is a more recent development aimed at very low-latency, high-quality compression for real-time streaming applications. While it does not focus on maximizing compression ratios, it offers visually lossless quality with minimal processing delay, making it ideal for professional video workflows, virtual reality (VR), and automotive applications.

Each of these JPEG extensions has been developed to address specific industry needs, offering enhancements in image quality, compression efficiency,

and flexibility. The variety of available extensions ensures that JPEG continues to evolve and meet the demands of modern imaging technologies.

## 2.5 Problems of compression

Data compression plays a crucial role in reducing the hardware resources and energy consumption required for various applications. However, it is important to note that compression remains one of the most resource-intensive processes due to its inherent complexity. This complexity is particularly evident in transformation-based compression techniques, which often involve computationally intensive operations compared to other image compression processes.

In response to this challenge, researchers have made significant efforts to mitigate the computational complexity associated with transformations. Various approaches, such as the development of multiplication-free transformations, have been explored in the literature [29, 30, 31, 32, 33, 34]. In the following sections, we will delve into some of these attempts to address and propose solutions for these complexities.

## 2.6 Conclusion

Image compression is essential for optimizing storage, transmission, and bandwidth efficiency across various applications. Despite its advantages, compression techniques, particularly transform-based methods, often face challenges due to their computational complexity and energy consumption. These issues are especially critical for resource-constrained systems like mobile and embedded devices.

In the following chapters, we will explore strategies to address these challenges, focusing on low-complexity, energy-efficient solutions to improve the performance of image compression in constrained environments.

# Part II

# Contributions

# Chapter 3

# Improving image encoding quality with a low-complexity DCT approximation using 14 additions

## Introduction

Image and video compression play crucial roles in various applications, including environment monitoring [35, 36], healthcare monitoring [37, 38], multimedia, and video surveillance [25, 39, 40]. However, the transformation step in most compression methods is both time and energy-intensive [41, 42]. In image compression applications like JPEG still compression [43] and video compression codecs [44], various transforms are utilized, with the discrete cosine transform (DCT) and its approximations being the most prevalent. The DCT is favored for its high energy compacting capability, offering a viable alternative to the statistically optimal Karhunen-Loeve transform (KLT) among linear transformations [45].

While numerous fast algorithms for exact DCT computation exist, their reliance on multiplication renders them less suitable for systems with limited resources [28] and real-time applications [46]. Consequently, multiplication-free approximations have been proposed in the literature to address this issue, aiming to reduce energy consumption while maintaining acceptable energy compaction [47, 48, 49, 34, 29, 30, 50, 51, 52, 53, 31].

To the best of our knowledge, two approximations of the 8-point DCT requiring only 14 additions, the MCB approximate DCT [54], and the Potluri approximate DCT ($\mathbf{P_{14}}$) [55], have been proposed in the literature. Addi-

tionally, there are also pruned DCT-like transformations for image and video compression that require less than or equal to 14 additions, such as the pruned BAS-2009 [56], and the pruned version of MCB [54] defined in [33], which require 14 and 10 additions, respectively.

The main objective of this chapter is to introduce a novel orthogonal DCT approximation with the lowest arithmetic complexity of 14 additions that offers better compression efficiency than the existing DCT approximations with the same complexity [54, 55]. The key contributions of our work are as follows:

- An algorithm that generates all integer candidate DCT approximations with 14 additions, based on the shape of the DCT matrix and the sign of its elements.
- A modified coding gain metric that better accounts for the quantization process in image compression.
- An optimization problem search based on the proposed metric, leading to a highly efficient 8-point DCT approximation that outperforms [54, 55].
- A pruned version of the proposed DCT approximation with only 10 additions achieves the highest compression efficiency compared to [33, 57].
- Assessmen

## 3.1 Background and related work

### 3.1.1 The MCB DCT approximation

The MCB [54] is derived from an approximate DCT matrix proposed in [29]. By replacing some elements of the original approximate DCT matrix with zeros, the MCB achieves a more computationally efficient representation. The MCB matrix can be defined as follows:

$$MCB = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{3.1}$$

The associated approximate DCT is given by $C_m = S_m \cdot MCB$, where

$$S_m = diag(\frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}) \tag{3.2}$$

### 3.1.2 The Potluri 2014 approximation ($\mathbf{P_{14}}$)

DCT approximation introduced in [55] aims to minimize the arithmetic complexity of the transformation matrix $T$ while maintaining orthogonality. This objective is achieved by optimizing the cost function $Cost(T)$, which represents the arithmetic complexity of $T$. The optimization problem can be formulated as $P_{14} = \text{ARG } \min_T\{Cost(T)\}$, where $\mathbf{P_{14}}$ denotes the desired matrix defined in Equation (3.3).

To obtain the best approximation, an exhaustive computational search was conducted to identify the eight most suitable candidate matrices from the results of the optimization problem. Ultimately, the selected matrix $\mathbf{P_{14}}$ was found to require only 14 additions and can be represented as follows:

$$P_{14} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \tag{3.3}$$

where the approximate DCT is expressed by $C_p = S_m \cdot P_{14}$ and $S_m$ is defined in (3.2).

## 3.2 Proposed DCT approximation [58]

### 3.2.1 8-points integer DCT

The proposed approximation $T_p$ is derived from the exact DCT kernel $C$ defined in (2.2). The approximation is subject to the following constraints:

1. $T_p \cdot T_p^T$ must be a diagonal matrix.
2. Elements of $T_p$ must be selected from the set $P = \{\pm 1, 0\}$.
3. The approximation must have the lowest possible arithmetic cost while maximizing the number of zero elements.

To satisfy these constraints, we set one value of $\gamma$ in each row of $C$ to 1 and the remaining values to 0.

The number of possible combinations for each row is calculated, resulting in 128 candidate matrices. Rows 1 and 5 are dependent only on $\gamma_3$ (2 rows and 1 value), so they have a single possible combination ($2^{1-1}$), while rows 2, 4, 6, and 8 have four different values of $\gamma$ (4 rows and 4 values of $\gamma$), and rows 3 and 7 have two (2 rows and 2 values of $\gamma$). Therefore, the number of combinations can be calculated as $4^{4-1} \times 2^{2-1} \times 2^{1-1} = 4^3 \times 2^1 \times 2^0 = 128$. This approach generates low-complexity approximations with most elements being null and only one set to 1.

Among the obtained matrices, only 12 matrices have satisfied the orthogonal property and have therefore been considered. These include the approximations denoted as $\mathbf{P_{14}}$ [55] and MCB [54], with the latter utilizing the highest value of $\gamma$ in each row and the former incorporating the first and second highest value of $\gamma$ for odd and even rows, respectively.

It should be noted that all 12 of the selected candidate matrices have the same traditional Cg [59]. This is due to the fact that all matrices result in the same transform coefficients, just in different sign and positions. The position of these coefficients is important because of the quantization process, which is a crucial aspect of image compression.

To obtain a more comprehensive evaluation, we have introduced a modified coding gain (mCg) that takes into account the quantization process defined in the JPEG standard. This allows the calculation of the transformed domain ($mR_y$) of the covariance matrix of $x$ ($R_x$), resulting in a more precise assessment of the performance of the approximation in terms of image compression quality.

$$mR_y = (C \cdot R_x \cdot C^T) \oslash Q, \tag{3.4}$$

with

$$Q = \begin{cases} Q_0 & \text{if } QF = 50 \\ round((Q_0 \cdot SF + 50) \div 100) & \text{otherwise ,} \end{cases}$$

$$Q_0 = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 65 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix},$$

$$SF = \begin{cases} 5000 \div QF & \text{if } QF < 50 \\ 200 - 2 \times QF & \text{if } QF > 50, \end{cases}$$

where $C$ is a transformation matrix, $\oslash$ denotes element-wise division, and QF is the quality factor controlling the compression rate. $R_x$ represents the covariance matrix of the signal $x$, with its elements calculated based on the exponentiated absolute difference of their corresponding indices, i.e., $\rho^{|i-j|}$, where $i$ and $j$ range from 1 to 8. In this study, the correlation factor $\rho$ is set to 0.95, which has been shown to be a reliable approximation for natural images in previous literature [59]. The mCg is computed using the same formula as the conventional coding gain, which is shown in equation (3.5). However, it is evaluated in the quantized domain $mR_y$ of $R_x$, which is defined in equation (3.4).

$$mC_g = 10log_{10}\frac{\frac{1}{N}\sum_{i=0}^{N-1}\sigma_{y_i}^2}{\left(\prod_{i=0}^{N-1}\sigma_{y_i}^2||f_i||^2\right)^{\frac{1}{N}}}, \tag{3.5}$$

where $N$ is the number of transform coefficients, $\sigma_{y_i}^2$ is the variance of the $i^{th}$ transform coefficient, which corresponds to the $i^{th}$ diagonal element of the matrix $mR_y$, and $||f_i||$ is the 2-norm of the $i^{th}$ basis function of the transform matrix.

Following a thorough evaluation of all 12 candidate matrices, we identified the optimal matrix, denoted as $T_p$, which exhibited the highest mCg. The determination of $T_p$ involved computing the mCg for each matrix using a QF of 90. After careful consideration, we selected matrix $T_p$, as specified in Equation (3.6), as the most suitable choice.

$$T_p = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{3.6}$$

The selected approximation, represented by the matrix $T_p$, employs the highest value of $\gamma$ for rows 1, 2, 3, 5, 7, and 8, and the lowest value of $\gamma$ for rows 4 and 6. Notably, matrix $T_p$ shares the odd rows of the MCB and $\mathbf{P_{14}}$ approximations, while displaying distinctive variations in the even rows. A comprehensive examination of these variations is provided in Section 3.2.2.

The derivation of the proposed approximation follows a systematic and iterative process, outlined in Algorithm 1. This algorithm offers a detailed, step-by-step procedure for achieving the proposed approximation, while ensuring low arithmetic complexity, orthogonality, and maximum mCg.

**Algorithm 1** The algorithm used for generation of $T_p$

**Require:** The exact DCT matrix $C$ of size $N$ defined in equation (2.2)
**Ensure:** An orthogonal matrix $T_p$ with the highest mCg.

```
max_Cg ← 0
M_c ← getAllCandidates(C)    ▷ Get all candidate matrices based on C
for T ∈ M_c do
    if T is orthogonal & mC_g(T) > max_Cg then    ▷ mCg is defined in equation (3.5)
        max_Cg ← mC_g(T)
        T_p ← T
    end if
end for
return T_p
```

Based on the theory of matrix polar decomposition [60], an adjustment matrix $S_p$ is sought to orthogonalize $T_p$. The orthogonalization matrix is obtained as follows:

$$S_p = \sqrt{(T_p \cdot T_p^T)^{-1}} \tag{3.7}$$

The computation described in equation (3.7) yields the same diagonal matrix as in [54, 55], defined as follows:

$$S_p = diag(\frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}) \tag{3.8}$$

It is important to note that $S_p$ is a diagonal matrix, which can be efficiently incorporated into the quantization process without adding any extra computational overhead. Thus, the orthogonal real elements kernel matrix $C_p$ can be defined as follows:

$$C_p = S_p \cdot T_p \tag{3.9}$$

### 3.2.2 Analysis of the proposed DCT approximation

The study compares three different DCT matrices: the exact DCT, MCB [54], and the proposed approximation. The MCB [54] and the proposed approximation were selected due to their high mCg values, indicating their effectiveness in preserving image information during compression. While all three

matrices are orthogonal, the quantization process affects the high-frequency elements differently. Therefore, arranging the matrices in ascending order of frequencies becomes crucial.

Each row in a transformation matrix acts as a linear phase Finite Impulse Response (FIR) filter. These filters are characterized by their zeros, which are plotted on the Z-plane (Fig. 3.1). In the case of DCT matrices, the zeros of all eight rows lie on the unit circle at various frequencies. As a result, each zero eliminates signals at its corresponding frequency entirely. The frequency bands between consecutive zeros determine the passband width or the presence of secondary lobes with attenuated gain. Understanding the

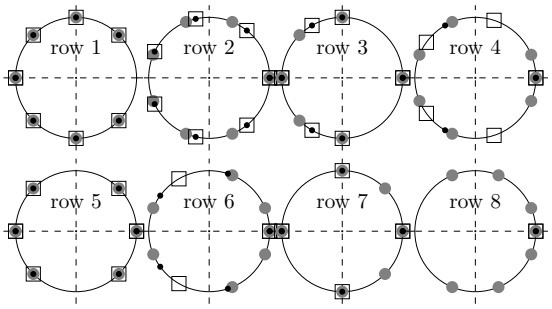



Figure 3.1: Z plane for the exact DCT, MCB and $T_p$. Big/gray points, small/black points and squares represent the zeros for: the exact DCT, the MCB and the proposed approximation, respectively.

An important characteristic of the exact DCT matrix is its gradual increase in bandwidth from low to high frequencies across successive rows. However, the MCB [54] deviates from this characteristic, especially in its even rows 4 and 6, resulting in a different frequency bandwidth order compared to the exact DCT. Conversely, the proposed approximation $T_p$ maintains the same frequency order as the exact DCT matrix, with the widest bandwidth shifting from low to high frequencies. This prioritizes the preservation of important data in the low-frequency range, as the quantization process predominantly affects high-frequency elements. Consequently, the proposed approximation may offer a more effective solution for retaining critical data while mitigating the impact of quantization on low-frequency components.

## 3.3 Performance assessments

This section assesses the performance of the proposed DCT approximation in comparison to existing approaches [54, 55]. We evaluate the Mean Squared Error (MSE) [59, 61] and total energy ($\epsilon$) [29] as similarity measures with

respect to the DCT. For coding performance evaluation, we use the coding gain Cg [59], modified coding gain mCg, and transform efficiency ($\eta$) [59]. Image quality assessment is conducted using PSNR [61] and Universal Quality Index (UQI) [62] absolute percentage error (APE) relative to the DCT, referred to as APE(UQI), for 10 retained coefficients.

Table 3.1: Performance assessment.

| Method | Cg[$dB$] | mCg[$dB$] | $\eta$ | MSE | $\epsilon$ | PSNR[$dB$] | APE(UQI) |
|---|---|---|---|---|---|---|---|
| Exact DCT | 8.826 | 12.366 | 93.99 | 0 | 0 | 30.6729 | 0 |
| $BAS_{\alpha=0}$ [49] | 7.912 | 11.4 | 85.65 | 0.071 | 26.86 | 28.9854 | 0.0767 |
| MCB [54] | 7.333 | 10.708 | 80.90 | 0.059 | 8.66 | 26.8539 | 0.2117 |
| $\mathbf{P_{14}}$ [55] | 7.333 | 10.702 | 80.90 | 0.079 | 11.31 | 27.4839 | 0.1748 |
| Proposed | 7.333 | 10.73 | 80.90 | 0.076 | 15.64 | 27.5454 | 0.163 |

The results presented in Table 3.1 indicate that the proposed DCT approximation and previous approaches [54, 55] demonstrate comparable coding performance when considering Cg and $\eta$. However, the proposed approximation exhibits slight improvements in terms of mCg, PSNR, and APE(UQI), leading to superior efficiency, as elaborated in Section 3.5.

When examining similarity measures, it is observed that the proposed approximation yields slightly higher MSE compared to MCB [54]. This can be attributed to MCB's utilization of the highest $\gamma$ values, while the proposed approximation employs the lowest values in rows 4 and 6. Conversely, $\mathbf{P_{14}}$ employs the highest and second-highest $\gamma$ values for even and odd rows of $C$, respectively, resulting in the highest MSE among all approximations. It is important to note that similarity measures may not fully define a transformation's suitability for image compression, as evidenced by the BAS series [47, 49]. Despite exhibiting high MSE and total error energy ($\epsilon$) compared to the exact DCT, the BAS series remains highly efficient in image compression.

Additionally, there exists a correlation between coding gain and transform efficiency ($\eta$), as both metrics are based on $R_y$. Therefore, the formula for $\eta$ can be modified to utilize $mR_y$ instead of $R_y$, resulting in distinct outcomes.

## 3.4 Proposed fast algorithms

### 3.4.1 $8 \times 8$ DCT Kernel

This section introduces a rapid algorithm utilizing a sparse matrix factorization technique for computing the proposed approximation $T_p$. By employing this method, the number of non-zero elements in the matrices is reduced,

leading to decreased computation requirements for computing the approximation. The approach involves multiplying several sparse matrices based on well-known butterfly structures [63], resulting in a significant reduction in computational cost and enhanced performance. Consequently, this method is well-suited for real-time applications and large-scale image processing tasks. The proposed approximation can be computed using the multiplication of the following sparse matrices: $T_p = P \cdot A_2 \cdot A_1 \cdot B$, where:

$$B = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \quad P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$A_1 = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad A_2 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

To better comprehend the data flow through the transformations, a SFG is provided in Fig. 3.2, showcasing the matrix decomposition of the proposed matrix $T_p$ alongside MCB [54] and $\mathbf{P_{14}}$. The structure of the proposed transformation mirrors that of the other two transformations, indicating similar hardware complexity in FPGA/ASIC implementation. However, the key difference lies in the order of the output $X$. Despite this similarity in architecture, it enables a direct comparison of the efficiency of the proposed transformation with other existing approximations. It's worth noting that the output $X_5$ of the proposed approximation is multiplied by -1. This operation can be executed without introducing additional arithmetic complexity by rearranging the order of the input $x_2$ to $x_5$, resulting in $-(x_5 - x_2) = -X_5$.

### 3.4.2 $4 \times 8$ Pruned Kernel

This section aims to conduct a thorough analysis of pruning schemes combined with approximate transforms, focusing on reducing the computational cost of the DCT within the framework of JPEG and JPEG-like coding and

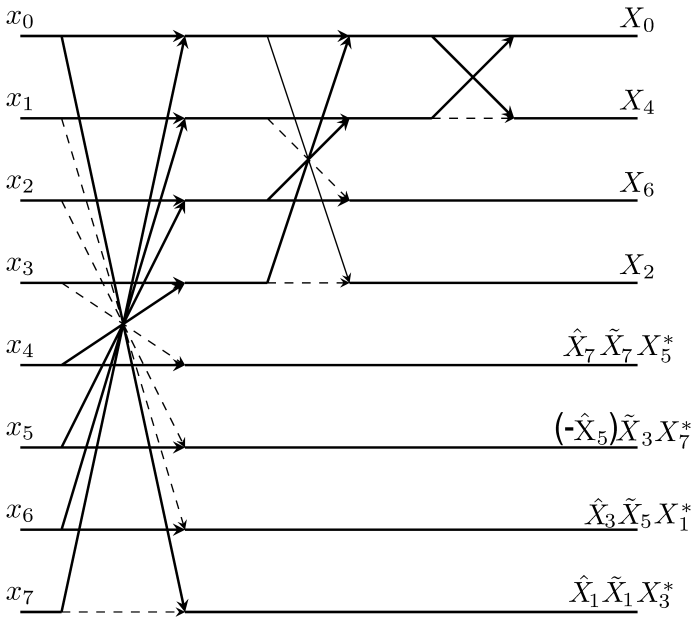



Figure 3.2: Signal flow graphs for $T_p$, MCB[54] and $\mathbf{P_{14}}$ [55]. Dashed lines represent multiplication by -1. $\hat{X}_m$, $\tilde{X}_m$ and $X_m^*$ outputs are for proposed, MCB and $\mathbf{P_{14}}$.

processing, while maintaining minimal impact on image quality. This endeavor is motivated by the increasing demand for high compression ratios in both image and video applications.

In JPEG image compression, the quantization process often eliminates high-frequency coefficients, optimizing computational efficiency. By retaining only low-frequency coefficients and discarding high-frequency ones, significant computational savings can be achieved. It's important to note that the arrangement of frequency elements in a transformation directly influences the quantization process. The proposed approximation organizes its frequency elements in a low-to-high frequency order, providing a distinct advantage over existing methods such as MCB [54] and $\mathbf{P_{14}}$ [55].

Taking these considerations into account, the following transformation is derived from the proposed approximation.

$$T_{p4} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{3.10}$$

Given the orthogonalization methods, a semi-orthogonal matrix can be obtained as follows:

$$C_4 = S_4 \cdot T_{p4}, \tag{3.11}$$

where $S_4 = diag(\frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}})$.

The proposed pruned approximation can be obtained by multiplying several sparse matrices. Let $T_{p4} = P^* \cdot A_2^* \cdot A_1^* \cdot B^*$, where:

$$B^* = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \qquad P^* = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$A_1^* = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad A_2^* = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The SFG for the fast algorithm of $T_{p4}$, which only requires 10 additions, positions it as a strong contender against Coutinho pruned of MCB (CMCB) [33] and $A_{20}$ [57]. Depicted in Figure 3.3, this graph illustrates the relationship between the input signal $x_n$ and the output signal $X_k$. Notably, the transform-domain components $X_4, \ldots, X_7$ are omitted from the graph and assumed to be zero.

### 3.4.3 Arithmetic complexity

Table 3.2: Arithmetic complexity comparison for 8-point DCT approximations.

| | 8-point 1D | | | Pruned 8-point 1D | | |
|---|---|---|---|---|---|---|
| Approximation | adds | b-shifts | Total | adds | b-shifts | Total |
| SDCT [64]/$Z_{19}$ [65] | 24 | 0 | 24 | 14 | 0 | 14 |
| CB [29]/$M_{16}$ [53] | 22 | 0 | 22 | 16 | 0 | 16 |
| $BAS_{\alpha=0}$/ - | 16 | 0 | 16 | - | - | - |
| MCB [54]/CMCB [33] | 14 | 0 | 14 | 10 | 0 | 10 |
| $\mathbf{P_{14}}$ [55]/$A_{20}$ [57] | 14 | 0 | 14 | 10 | 0 | 10 |
| Proposed $T_p$/$T_{p4}$ | 14 | 0 | 14 | 10 | 0 | 10 |

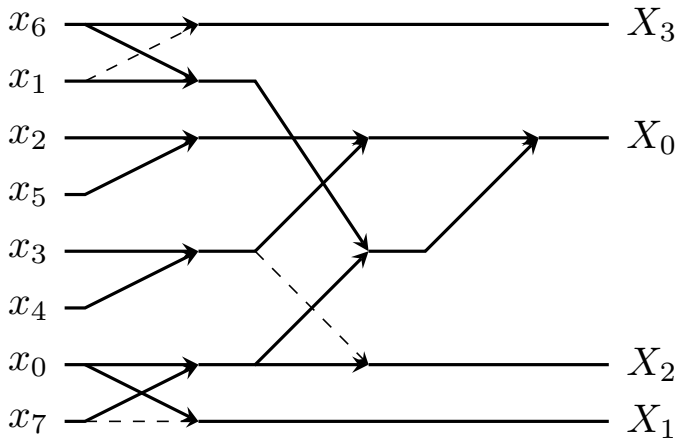



Figure 3.3: Signal flow graph for pruned $T_p$. Dashed lines represent multiplication by -1.

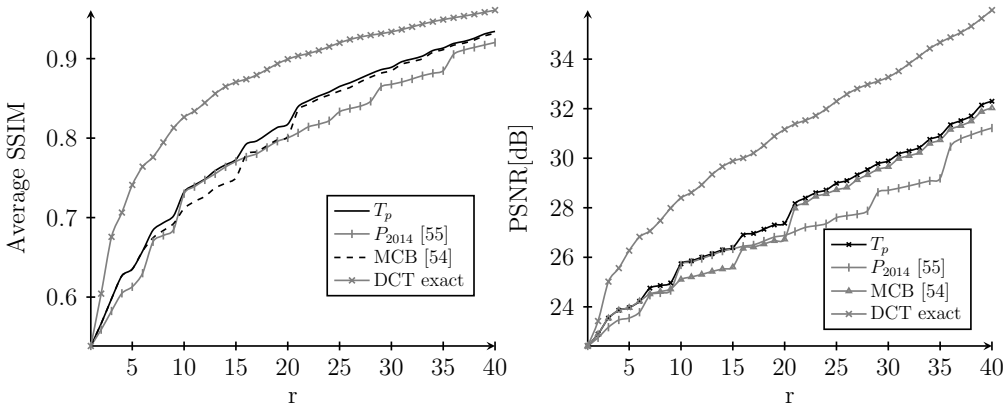
Table 3.2 offers a detailed comparison of the computational complexity associated with various DCT approximations, including the proposed method and other state-of-the-art techniques. The table enumerates the number of additions and bit-shifts required for each approximation, providing valuable insights into their computational requirements. The findings reveal that both the proposed DCT approximation and its pruned variant demonstrate comparable computational costs to existing approaches such as [54, 55, 33, 57]. Notably, these proposed methods exhibit lower complexity compared to other alternatives. Despite sharing similar arithmetic complexities with [54, 55, 33, 57] in terms of additions and bit-shifts, the proposed DCT approximation and its pruned version deliver superior image quality. This enhanced performance makes them particularly well-suited for applications requiring efficient and rapid signal processing.

## 3.5 Image compression application

The proposed algorithm and transformations, as discussed in prior research [54, 55, 33, 57], were evaluated within the context of JPEG-like and JPEG compression. These techniques, exhibiting equivalent arithmetic complexity as outlined in Table 3.2, were subjected to experimentation. Initially, a dataset comprising 47 greyscale images was obtained from a publicly available image repository [66].

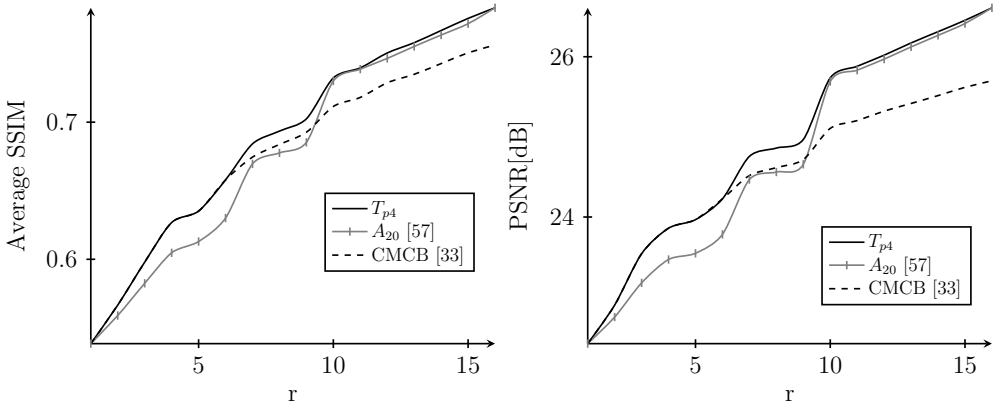
### 3.5.1 JPEG-like

Each image is partitioned into $8 \times 8$ blocks and subjected to a 2-D transform, yielding 64 coefficients arranged in a zigzag pattern [43]. From each block, only the first $r$ coefficients are retained, with $1 \leq r \leq 40$, while the rest are discarded. The decompressed images resulting from the inverse 2-D transforms are then compared to the original images. Image quality assessment is conducted using metrics such as (1) PSNR [61] and (2) SSIM [67].




(a) Average SSIM vs r (b) Average PSNR vs r

Figure 3.4: Image quality measures for several compression ratios (r).




(a) Average SSIM vs r (b) Average PSNR vs r

Figure 3.5: Image quality measures for several compression ratios (r) of the pruned kernels.

The findings reveal the superior performance of our proposed DCT approximation compared to MCB [54] and $\mathbf{P_{14}}$ [55] in terms of image quality,

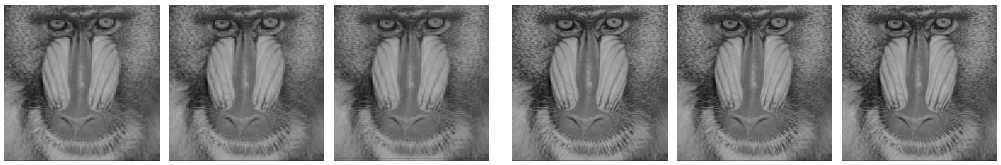
as demonstrated by SSIM, and PSNR metrics in Fig. 3.5. This advantage holds true for all values of retained coefficients $r$, except for $r < 7$ where MCB [54] shows comparable results. On average, our proposed approximation achieves a PSNR improvement of 0.3734 dB and 1.0911 dB over MCB and $\mathbf{P_{14}}$, respectively.

Moreover, the pruned variant of our proposed DCT approximation surpasses the CMCB [33] and $\mathbf{P_{14}}$ [55] ($A_{20}$ [57]), as demonstrated in Fig. 3.4. Specifically, our pruned matrix exhibits image quality comparable to CMCB [33] for $r < 7$, and outperforms it for $r \geq 7$. Additionally, the pruned DCT approximation, $T_{p4}$, outperforms $A_{20}$ [57], showing a significant improvement in quality for $r < 10$ and a slight advantage for $r \geq 10$. These results are consistent across all image samples used in the experiment and can be generalized to other datasets.

The zig-zag ordering ensures a sorting of coefficients from low to high frequency if the transform matrix adheres to this condition. Equation (3.12) delineates this order, emphasizing discrepancies between the proposed transform and MCB [54] with a shaded area. These discrepancies become apparent from coefficient 7 ($r7$), as illustrated in Figures 3.5a and 3.5b, where the proposed transform exhibits a slightly superior performance starting from $r = 7$. This trend persists until $r = 61$, beyond which the proposed and MCB [54] show similar performance for $r \geq 62$. This improvement can be anticipated because row/column 4 in the proposed transform corresponds to a lower frequency compared to the same row/column in MCB [54], suggesting a higher emphasis on low-frequency components in the proposed transform.

$$Y = \begin{pmatrix} r1 & r2 & r6 & r7 & r15 & r16 & r28 & r29 \\ r3 & r5 & r8 & r14 & r17 & r27 & r30 & r43 \\ r4 & r9 & r13 & r18 & r26 & r31 & r42 & r44 \\ r10 & r12 & r19 & r25 & r32 & r41 & r45 & r54 \\ r11 & r20 & r24 & r33 & r40 & r46 & r53 & r55 \\ r21 & r23 & r34 & r39 & r47 & r52 & r56 & r61 \\ r22 & r35 & r38 & r48 & r51 & r57 & r60 & r62 \\ r36 & r37 & r49 & r50 & r58 & r59 & r63 & r64 \end{pmatrix} \tag{3.12}$$

To further support this theory, three distinct image types from another dataset [66] were chosen: 'Pepper', 'Baboon', and 'Lena', each representing very low, high, and medium frequencies, respectively (Fig. 3.6). These images were tested before, after, and between the breakpoints ($r < 7$, $r > 61$, and $7 \leq r \leq 61$). Figs. 3.6j1 and 3.6j3 depict the reconstructed images for $r < 7$ and $r > 61$, respectively. The results indicate that the proposed approximation and MCB [54] exhibit comparable quality in all cases, with both transformations outperforming $\mathbf{P_{14}}$ [55]. Conversely, between break-

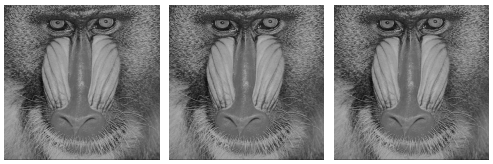


(a1) $T_p$ PSNR = 22.25dB (b1) MCB PSNR = 22.25dB (c1) $P_{2014}$ PSNR = 22.00dB (a2) $T_p$ PSNR = 23.77dB (b2) MCB PSNR = 23.61dB (c2) $P_{2014}$ PSNR = 23.76dB

(d1) $T_p$ PSNR = 27.04dB (e1) MCB PSNR = 27.04dB (f1) $P_{2014}$ PSNR = 25.86dB (d2) $T_p$ PSNR = 30.22dB (e2) MCB PSNR = 28.63dB (f2) $P_{2014}$ PSNR = 30.09dB

(g1) $T_p$ PSNR = 28.5dB (h1) MCB PSNR = 28.5dB (i1) $P_{2014}$ PSNR = 27.77dB (g2) $T_p$ PSNR = 31.25dB (h2) MCB PSNR = 30.01dB (i2) $P_{2014}$ PSNR = 31.19dB

(j1) Reconstructed images with $r = 6$. (j2) Reconstructed images with $r = 15$.



(a3) $T_p$ PSNR = 42.20dB (b3) MCB PSNR = 42.20dB (c3) $P_{2014}$ PSNR = 38.40dB

(d3) $T_p$ PSNR = 49.96dB (e3) MCB PSNR = 49.96dB (f3) $P_{2014}$ PSNR = 46.63dB

(g3) $T_p$ PSNR = 52.97dB (h3) MCB PSNR = 52.97dB (i3) $P_{2014}$ PSNR = 46.07dB

(j3) Reconstructed images with $r = 62$.

Figure 3.6: Reconstructed images for each of breakpoints

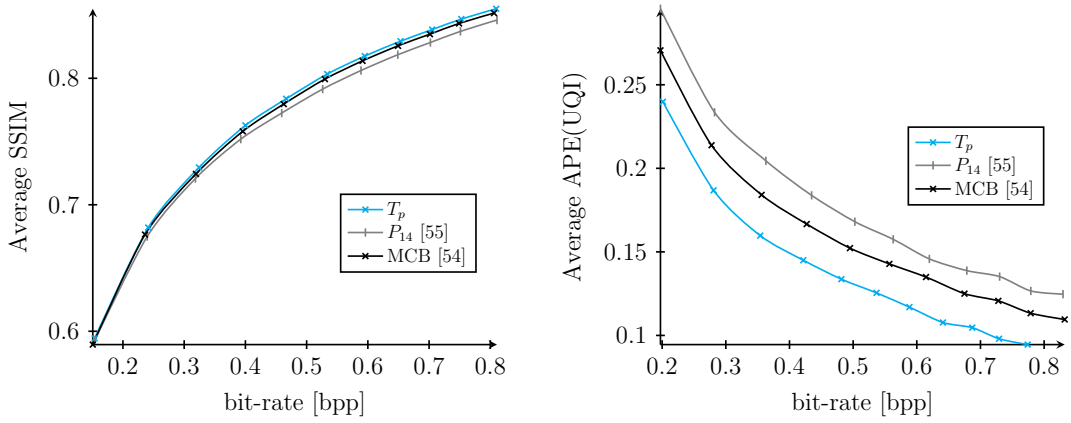
points where $7 \leq r \leq 61$ (Fig. 3.6j2), the proposed approximation surpasses MCB [54] and $\mathbf{P_{14}}$ [55] in terms of quality. This demonstrates that our approximation performs exceptionally well across various scenarios, such as grass and animals and birds feathers (high frequencies), sky and water (low frequencies), and everything in between.

### 3.5.2 JPEG

In the preceding subsection, it was demonstrated that the proposed approximation performs comparably to MCB [54] in terms of reconstructed image quality when the number of retained coefficients ($r$) is below 7. However, for $r$ values greater than or equal to 7, the proposed approximation exhibits superior performance in terms of image quality compared to other approximations. It is noteworthy that despite achieving a high compression rate, which is often a key consideration, the proposed approximation does not yield higher image quality when the rate exceeds 90%. These findings were obtained within the framework of JPEG-like compression, where quantization and Huffman coding were not utilized. To comprehensively evaluate the efficacy of the proposed approximation and its impact on image quality, further testing is warranted within a complete JPEG compression chain incorporating quantization and Huffman coding.

The proposed DCT approximation, $T_p$, has been compared against other existing approximations in terms of image quality metrics including SSIM, APE(UQI), and PSNR, as presented in Fig. 3.7. The results indicate that $T_p$ consistently outperforms other approximations and has the best overall performance. Specifically, $T_p$ outperforms $\mathbf{P_{14}}$ [55] and has a slight gain compared to MCB [54]. Furthermore, the pruned version of each approximation has been evaluated as shown in Fig. 3.8. The results indicate that the pruned version of $T_p$ ($T_{p4}$) surpasses the other approximations and demonstrates improved gains compared to CMCB [33] and a small gain compared to the pruned version of $\mathbf{P_{14}}$ ($A_{20}$ [57]), which becomes the second-best performer. On the other hand, the pruned version of MCB (CMCB [33]) shows a significant loss in quality of around 1.5 dB.

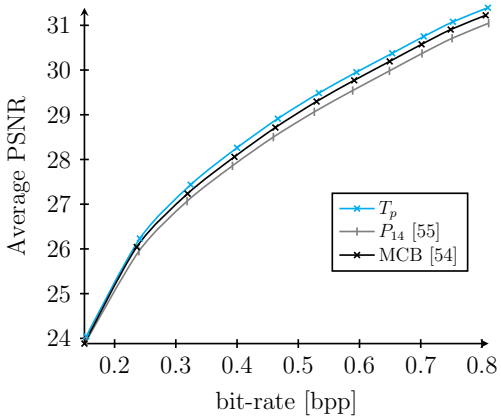
Thus, the proposed transform offers a distinct advantage for target-driven JPEG compression applications [68, 69, 70], where region-adaptive compression ratios (CR) are employed to optimize performance. In this approach, critical regions can be compressed with lower compression ratios (below 90%) using $T_p$, preserving high image quality, while less critical regions are compressed with higher compression ratios (above 90%) using the pruned version $T_{p4}$ to reduce arithmetic complexity. This strategy provides optimal performance, ensuring that the regions of interest (ROI) retain high quality while




(a) Average SSIM vs bit-rate

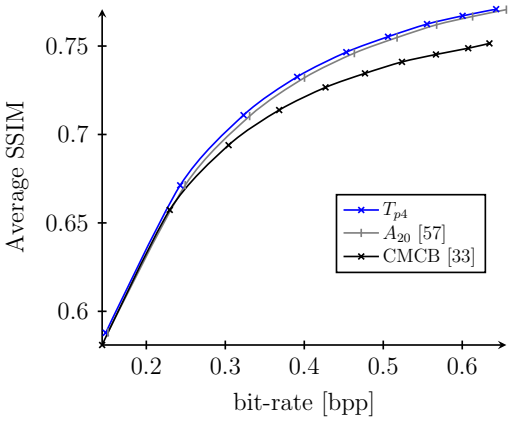



(b) Average APE(UQI) vs bit-rate

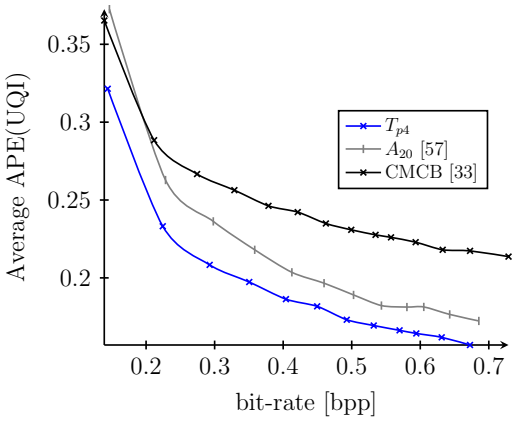



(c) Average PSNR vs bit-rate.

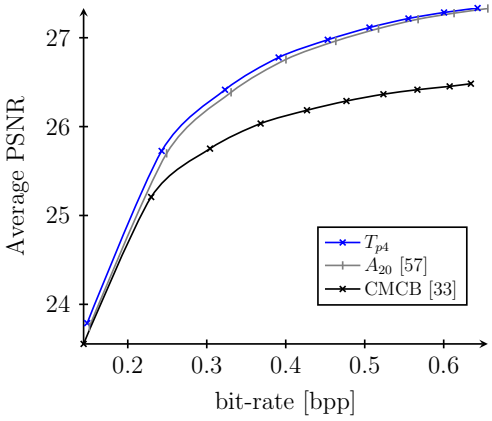
Figure 3.7: Image quality measures for several compression ratios (bit rate).




(a) Average SSIM vs bit-rate for pruned matrices.



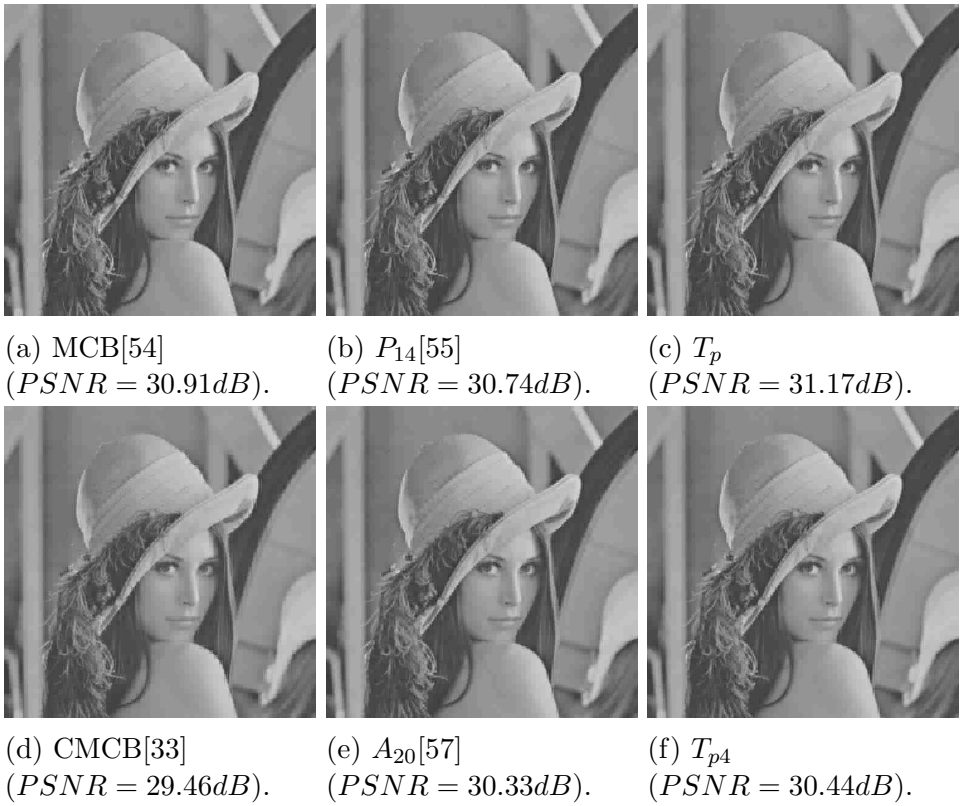

(b) Average APE(UQI) vs bit-rate for pruned matrices.




(c) Average PSNR vs bit-rate for pruned matrices.

Figure 3.8: Image quality measures of the pruned versions for several compression ratios (bit rate).

utilizing a single architecture for both tasks.



(a) MCB[54] ($PSNR = 30.91dB$). (b) $P_{14}$[55] ($PSNR = 30.74dB$). (c) $T_p$ ($PSNR = 31.17dB$).

(d) CMCB[33] ($PSNR = 29.46dB$). (e) $A_{20}$[57] ($PSNR = 30.33dB$). (f) $T_{p4}$ ($PSNR = 30.44dB$).

Figure 3.9: Reconstructed image of 'Lena' for bitrate=0.3bpp.

Moreover, the proposed approximation yields impressive results, at a compression rate of 96.25% (bit-rate of 0.3 bpp), as detailed in Table 3.3 and depicted in Fig. 3.9. The proposed approximation consistently outperforms both the MCB and $\mathbf{P_{14}}$ approximations in terms of image quality.

The tested images, which contained a variety of frequency components, demonstrated that the proposed approximation significantly outperformed both the MCB and $P_{14}$ approximations in terms of image quality. For the 'Lena' image, the quality improvements were 0.26 dB and 0.43 dB over MCB and $P_{14}$, respectively. Similarly, for the 'Pepper' image, the gains were 0.2 dB and 0.45 dB, while the 'Baboon' image showed enhancements of 0.05 dB and 0.11 dB compared to MCB and $P_{14}$, respectively. These results highlight the suitability of the proposed approximation for embedded systems and other applications that require high compression rates without significant loss in image quality.

These findings underscore the suitability of the proposed approximation for embedded systems and other applications requiring high compression

rates without compromising image quality.

Table 3.3: Quality of reconstructed images (PSNR[dB]) at a bit-rate of 0.3 bpp

| Method | Lena | Pepper | Baboon |
|---|---|---|---|
| MCB[54] | 30.91 | 29.99 | 22.57 |
| $P_{14}$[55] | 30.74 | 29.72 | 22.63 |
| $T_p$ | 31.17 | 30.19 | 22.68 |
| CMCB[33] | 29.46 | 27.88 | 22.42 |
| $A_{20}$[57] | 30.33 | 28.83 | 22.46 |
| $T_{p4}$ | 30.44 | 28.98 | 22.47 |

The proposed pruned approximation, denoted as $T_{p4}$, demonstrates superior performance compared to the MCB pruned matrix [33]. This is reflected in an image quality enhancement of 1.1 dB for 'Pepper', 0.98 dB for 'Lena', and 0.05 dB for 'Baboon'. These results indicate a consistent improvement over the MCB approximation [54] and $P_{14}$ [55], with $T_{p4}$ maintaining its advantage in performance metrics.

Despite these improvements, a marginal reduction in quality is noted when comparing $T_{p4}$ to $P_{14}$, particularly with decreases of 0.1 dB for 'Baboon', 0.74 dB for 'Pepper', and 0.3 dB for 'Lena'. Similarly, when evaluated against MCB, the proposed approximation exhibits a loss of 0.16 dB for 'Pepper', 1.01 dB for 'Lena', and 0.47 dB for 'Baboon'.

A significant advantage of $T_{p4}$ is its reduced computational complexity, requiring only 10 additions, compared to the 14 additions necessary for both MCB and $P_{14}$. This reduction in arithmetic complexity presents a notable improvement in resource efficiency, making $T_{p4}$ a compelling candidate for applications in resource-constrained environments.

## 3.6 Conclusion

This chapter introduced a novel DCT approximation that demonstrates superior performance compared to previous state-of-the-art 14-addition approximations, namely MCB [54] and $\mathbf{P_{14}}$ [55], particularly in terms of image quality. The proposed approximation distinguishes itself through its low arithmetic complexity, similar hardware implementation requirements, and enhanced compression outcomes, making it particularly suitable for embedded systems. This is especially relevant for embedded monitoring applications, where energy efficiency and limited hardware resources are critical.

Extensive testing across various image datasets revealed the effectiveness of the proposed approximation, as indicated by significant improvements in PSNR values. The introduction of the modified coding gain (mCg) metric further corroborates the efficacy of this new approach, showing a clear correlation between higher mCg values and better image quality. Additionally, the pruned version of the approximation, $T_{p4}$, also yielded promising results, outperforming existing alternatives such as CMCB [33] and $A_{20}$ [57] in terms of image quality.

Given its capabilities, the proposed DCT approximation holds considerable potential for embedded systems and applications that require high compression rates while maintaining satisfactory image quality. Future research will focus on refining existing transform approximations using the proposed algorithm, exploring more efficient hardware implementations, and investigating the possibility of a single architecture with multiple pruned versions for region-of-interest applications. Additionally, efforts will be directed toward further enhancing image quality by developing a specialized quantization matrix tailored to the proposed approximation. Another promising direction for future work is to explore the DTT as a complementary or alternative approach, potentially yielding further improvements in compression efficiency and complexity reduction in embedded systems.

# Chapter 4

# Improved DTT approximations for efficient image compression

## Introduction

Discrete orthogonal transforms play a pivotal role in numerous signal processing applications, including image and video compression, pattern recognition, digital watermarking, and security. Among these transforms, the KLT is mathematically optimal for data decorrelation and energy compaction. However, its data-dependent nature limits the development of fast algorithms, hindering its practical implementation. Consequently, the DCT has emerged as an attractive approximation to the KLT due to its data-independent computation and the availability of several fast algorithms [63, 71, 72, 73]. As a result, the DCT has been extensively adopted in various standards and signal processing techniques, such as JPEG image compression [43], video coding [74, 44, 75], and digital watermarking [76, 77].

Despite the existence of efficient algorithms, the computational complexity of the DCT remains a significant challenge, particularly in resource-constrained embedded systems [78, 39] and real-time applications [39]. To address this issue, transform approximations have been proposed as an alternative approach, aiming to reduce the computational burden while maintaining acceptable performance. These approximations can be broadly classified into two categories: (i) substituting the original transform with a low-accuracy approximation [50, 79, 34, 58, 80], and (ii) pruning or truncating coefficients that are likely to be negligible or zero [81, 82, 83, 84, 85].

In recent years, the DTT has gained increasing attention as an alternative to the DCT for signal processing applications. The DTT, derived from discrete Tchebichef polynomials, exhibits similar properties to the DCT, such

as decorrelation and energy compaction capabilities. In some cases, the DTT has demonstrated superior performance compared to the DCT in image compression tasks [86, 27, 87]. The DTT has been successfully applied in various domains, including image analysis [88], security [89, 23], digital watermarking [90], pattern recognition [91], video interpolation [92], and image/video coding [93, 82].

However, despite its promising properties, the computational complexity of the DTT remains a significant challenge, particularly for the widely used 8-point DTT. While numerous fast algorithms have been developed for the DCT, only a limited number of approaches have been proposed to reduce the complexity of the DTT [94, 32, 95]. These existing DTT approximations suffer from various shortcomings, such as neglecting the deviation from orthogonality as an optimization criterion [94, 32], employing distinct algorithms for forward and inverse transformations [94], or exhibiting deviations in the low-frequency region [95], which can adversely affect compression performance.

To address these limitations, the present work proposes novel approaches to approximating the DTT, resulting in reduced computational complexity and improved compression efficiency compared to the existing approximations [94, 32, 95]. The key contributions of this work are as follows:

- A critical analysis of previous DTT approximations [94, 32, 95], highlighting their strengths and weaknesses.
- Introduction of a modified deviation metric that considers the location of deviations, leading to optimized approximations with enhanced compression performance.
- Investigation of using the inverse matrix, in addition to the transpose matrix, for computing the 2D DTT, enabling a comprehensive evaluation of approximation accuracy.
- Development of efficient fast algorithms for computing the proposed DTT approximations, further reducing their computational complexity.
- Evaluation of the approximations' performance using various metrics, including Cg, transform efficiency ($\eta$), and MSE, to assess their compression capabilities.
- Consideration of hardware resource utilization, processing speed, and energy consumption in FPGA implementations, providing insights into their practical applicability.

- Implementation of the proposed DTT approximations in a JPEG-like compression application, demonstrating their superiority over existing approximations through comparative analysis.

The proposed work aims to address the computational complexity challenges of the DTT while maintaining competitive compression performance, enabling its wider adoption in resource-constrained signal processing applications.

## 4.1 Review of fast algorithms for the 8-point DTT

As previously stated in the introduction, fast algorithms for the DTT are scarce in the signal processing literature compared to the DCT. There is a significant need for more proposals of DTT approximations. To the best of our knowledge, there exist only three fast algorithms in the literature for computing an 8-point DTT. The first one was proposed in [86] and provides an exact DTT using the kernel described in (4.9). The other two algorithms [94, 32] compute DTT approximations, with kernels given in (4.11) and (4.12) for approximation [94] and (4.14) for approximation [32]. It is worth mentioning that the works in [81, 82] pertain to the pruned version of the DTT, while those in [83, 96, 97] concentrate on hardware optimizations, which are not within the scope of this current work.

### 4.1.1 Exact DTT

The DTT is an orthogonal transformation derived from the discrete Tchebichef polynomials [98, 88]. The $k^{th}$ order of the polynomials is defined as follows:

$$t_{k,n} = P_1(k) \cdot P_2(k,n), \tag{4.1}$$

with:

$$P_1(k) = \sqrt{\frac{(2k+1)(N-k-1)!}{(N+k)!}} \cdot (1-N)_k,$$

$$P_2(k,n) =_3 F_2(-k,-n,1+k;1,1-N;1),$$

$$k,n = 0,1,\ldots,N-1,$$

$${}_3F_2(a_1,a2,a3;b1,b2;z) = \sum_{k=0}^{\infty} \frac{(a_1)_k(a_2)_k(a_3)_k}{(b1)_k(b2)_k} \cdot \frac{z^k}{k!},$$

where ${}_3F_2$ is the hyper-geometric function and $(a)_k$ is the ascending factorial defined in (4.2).

$$(a)_k = a(a+1)\dots(a+k-1), k \geq 1 \text{ and } (a)_0 = 1 \tag{4.2}$$

The DTT is derived from the discrete Tchebichef polynomials, which exhibit properties such as orthogonality, recurrence, and normalization. In the DTT domain, also known as Discrete Orthogonal Moments (DOMs), moments can be calculated by the use of discrete orthogonal polynomials. The transformed 2D sequence, $Y_{nm}$, which represents the set of discrete orthogonal 2D moments, is obtained from the input data sequence $X(x, y)$, are described in [99].

$$Y_{nm} = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} t_{n,x}t_{m,y}X(x,y); n, m = 0, 1, \dots, N-1 \tag{4.3}$$

The orthogonality property leads to the following inverse 2D transform.

$$X(x,y) = \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} Y_{nm}t_{n,x}t_{m,y}; x, y = 0, 1, \dots, N-1 \tag{4.4}$$

The 2-D DTT of an input data sequence $X$, with size $N \times N$, can be expressed in the matrix form, as described by (4.5)

$$Y = T_N \cdot X \cdot T_N^T, \tag{4.5}$$

where the N-point real matrix $T_N$ of the DTT kernel can be obtained by :

$$T_N = \begin{pmatrix} t_{0,0} & t_{0,1} & \cdots & t_{0,N-1} \\ t_{1,0} & t_{1,1} & \cdots & t_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ t_{N-1,0} & t_{N-1,1} & \dots & t_{N-1,N-1} \end{pmatrix} \tag{4.6}$$

$X$ is the input data of size $N \times N$ and $Y$ is the transform-domain coefficients of $X$. The inverse procedure can recover the original intensity distribution $X$ as follows:

$$X = T_N^T \cdot Y \cdot T_N \tag{4.7}$$

The matrix $T_N$ is orthogonal, which means that its transpose is equal to its inverse. Then, $T_N \cdot T_N^T = I_N$, where $I_N$ is the identity matrix of size $N \times N$. The real 8-point DTT kernel matrix used in [86] can be described by the product of the diagonal matrix $D$ and the integer-entry matrix $T_0$, as described in (4.8)

$$T_8 = D_0 \cdot T_0, \tag{4.8}$$

where

$$T_0 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -7 & -5 & -3 & -1 & 1 & 3 & 5 & 7 \\ 7 & 1 & -3 & -5 & -5 & -3 & 1 & 7 \\ -7 & 5 & 7 & 3 & -3 & -7 & -5 & 7 \\ 7 & -13 & -3 & 9 & 9 & -3 & -13 & 7 \\ -7 & 23 & -17 & -15 & 15 & 17 & -23 & 7 \\ 1 & -5 & 9 & -5 & -5 & 9 & -5 & 1 \\ -1 & 7 & -21 & 35 & -35 & 21 & -7 & 1 \end{pmatrix}, \tag{4.9}$$

$$D_0 = \frac{1}{2} \cdot diag(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{42}}, \frac{1}{\sqrt{42}}, \frac{1}{\sqrt{66}}, \frac{1}{\sqrt{154}}, \frac{1}{\sqrt{546}}, \frac{1}{\sqrt{66}}, \frac{1}{\sqrt{858}}) \tag{4.10}$$

The scaling matrix $D_0$ is merged into the quantization step and does not affect the complexity of the transformation step. As a result, only the integer-entry matrix $T_0$ is employed to improve the computational efficiency through the implementation of a multiplication-free algorithm. This technique utilizes the dyadic decomposition of the integer DTT coefficients. For instance, when the input data $x$ needs to be multiplied by a constant $a$, representing an element of the DTT matrix kernel, this constant is written as a sum of power of 2. Consequently, the multiplication of $x$ by this constant can be achieved through additions and bit shifting, where n-bit shifts equate to a multiplication by $2^n$.

In [86], a fast algorithm of the integer DTT matrix $T_0 = D_0^{-1} \cdot T_8$ (where $\cdot$ denotes matrix multiplication and $T_8$ represents the real DTT kernel) was proposed. This algorithm requires 44 additions and 29 bit-shifting operations, which is considered complex compared to its subsequent discrete transform approximations that require fewer than 30 operations [94, 32].

### 4.1.2 DTT approximations

A DTT approximation aiming at reducing the number of operations was proposed in [94]. But, the resulting algorithm remains complex with 20 operations required for the forward 1D transform and 37 operations for the inverse 1D transform, due to its non-orthogonality. The kernel used for this

approximation, referred to as $O^*$, is presented in (4.11).

$$O_{15}^* = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ -1 & 1 & 1 & 0 & 0 & -1 & -1 & 1 \\ 0 & -1 & 0 & 1 & 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & -1 & 1 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 \end{pmatrix} \tag{4.11}$$

The inverse transform kernel can be computed using $(O_{15}^*)^{-1} = O_{15} \cdot D_{15}$, where $D_{15}$ is the diagonal matrix with the elements $diag(\frac{1}{8}, \frac{1}{10}, \frac{1}{8}, \frac{1}{10}, \frac{1}{4}, \frac{1}{10}, \frac{1}{8}, \frac{1}{10})$ and

$$O_{15} = \begin{pmatrix} 1 & -3 & 3 & -2 & 1 & -1 & -1 & -1 \\ 1 & -2 & -1 & 2 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & -2 & 3 & -2 \\ 1 & -1 & -1 & 1 & 1 & -2 & -1 & 3 \\ 1 & 1 & -1 & -1 & 1 & 2 & -1 & -3 \\ 1 & 1 & -1 & -1 & -1 & 2 & 3 & 2 \\ 1 & 2 & -1 & -2 & -1 & -1 & -1 & -1 \\ 1 & 3 & 3 & 2 & 1 & 1 & -1 & 1 \end{pmatrix} \tag{4.12}$$

Therefore, the 2-D DTT domain $Y$ of the input data $X$ can be expressed by:

$$Y = O_{15}^* \cdot X \cdot (O_{15}^*)^{-1} = O_{15}^* \cdot X \cdot O_{15} \cdot D_{15} \tag{4.13}$$

In [32], another approximation, referred to as $O_{16}$, has been proposed and has demonstrated better results in terms of both quality and complexity compared to the previous approximation [94]. The $O_{16}$ is quasi-orthogonal and requires the same number of operations, 30, for both the forward and inverse transform. The kernel matrix for this transformation can be summarized as follows:

$$O_{16} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & -1 & 0 & 0 & 1 & 1 & 2 \\ 2 & 0 & -1 & -1 & -1 & -1 & 0 & 2 \\ -2 & 1 & 2 & 1 & -1 & -2 & -1 & 2 \\ 1 & -2 & 0 & 1 & 1 & 0 & -2 & 1 \\ -1 & 2 & -1 & -1 & 1 & 1 & -2 & 1 \\ 0 & -1 & 2 & -1 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -2 & 1 & 0 & 0 \end{pmatrix}, \tag{4.14}$$

where its scaling matrix $S_{16}$ is

$$S_{16} = diag(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{14}}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{10}})$$

### 4.1.3 Analysis of the previous DTT approximations

Orthogonality is a fundamental property in numerous signal processing applications, particularly in the context of image compression, owing to its numerous advantages such as the ability to use the transpose matrix instead of the inverse matrix, identical arithmetic complexity for both forward and inverse 1D transforms, the utilization of the same hardware architecture for both transforms, and reversibility, where the recovered data matches the input data, as expressed by $T \cdot T^{\top} = I$, where $I$ is the identity matrix.

However, the use of approximations of the exact DTT kernel may result in a reduction in these benefits. For instance, the DTT approximation $O_{15}$ [94] exhibits a relatively significant deviation from orthogonality, rendering it unsuitable for use in the reconstruction process. The need for the inverse matrix in the reconstruction process owing to the low Cg of $O_{15}$ and its deviation from orthogonality leads to the need for two distinct hardware architectures and a higher arithmetic complexity.

The DTT approximation $O_{16}$ presented in [32] offers a lower deviation from orthogonality, rendering it quasi-orthogonal. This approximation can utilize the transpose instead of the inverse matrix in the reconstruction process, resulting in equal arithmetic complexity for both transforms and the utilization of the same hardware architecture. However, it is crucial to note that using the transpose of a non-orthogonal matrix may result in certain issues, such as $T^{-1} \neq T^{\top}$ and $T \cdot T^{\top} \neq I$.

## 4.2 First Proposition ($T_{p1}$) [95]

The aim of this section is to derive a low-complexity approximation of the DTT, which is based on the exact DTT. The methodology involves generating a set of parametric integer matrices and identifying the best candidate based on orthogonality, coding performance, and proximity to the exact DTT. In the literature on the DCT, several notable approximations have been proposed, such as in [50, 100, 30]. Typically, an approximate transform behaves similarly to the exact matrix, $C$, with respect to specified metrics. The scale-and-round approach [30] provides a method for deriving a DCT approximation by using the following relationship:

$$T = \text{round}(\alpha \cdot C), \tag{4.15}$$

where round() is a rounding function that rounds to the nearest integer, $\alpha$ is a real parameter, and $C$ is the exact 8 by 8 DCT matrix. A similar methodology to the scale-and-round approach was used in [94, 32] to derive DTT approximations.

### 4.2.1 Parametric $8 \times 8$ integer matrices

The proposed DTT approximation is based on the exact DTT [86] and uses the same scale-and-round approach as seen in the previous DTT approximations [94, 32]. However, it differs from [94, 32] by finding an optimal scaling factor, $\alpha_i$, for each row of the matrix $T_0$, instead of using a single scaling factor for all rows. This methodology has two benefits. Firstly, the candidate matrix set becomes larger compared to using a single scaling factor, leading to a better solution. Secondly, the rows of the DTT matrix (i.e., basis vectors) have a large dynamic range in comparison to the DCT, making the round function less precise. Thus, the round error introduced will not be evenly distributed. By using a different parameter for each row, this phenomenon can be minimized.

The initial step in deriving the proposed DTT approximation is to normalize the rows of the exact DTT. This is done by dividing each row by its absolute maximum value, resulting in values between -1 and 1. This is equivalent to a left-multiplication by a scaling diagonal matrix, as shown in (4.16):

$$T^* = D_0 \cdot T_0, \tag{4.16}$$

where $T^*$ is a real-valued matrix on which our DTT approximation will be based, and $T_0$ is the integer DTT matrix defined in (4.9), and

$$D_0 = diag(1, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{13}, \frac{1}{23}, \frac{1}{9}, \frac{1}{35}) \tag{4.17}$$

Our aim is to develop a low-complexity approximation of the DTT matrix, with elements from the set $P_2 = \{\pm 2, \pm 1, 0\}$, using the round function. This set of integers is suitable for hardware implementation with low energy consumption as it requires only addition and bit-shifting operations. To achieve this, we can generate a set of parametric integer matrices by applying the round function to the product of the diagonal matrix, $\alpha$, and the normalized DTT matrix, $D_0 \cdot T_0$

$$T_{p1}(\alpha) = round(\alpha \cdot D_0 \cdot T_0), \tag{4.18}$$

where $\alpha$ is a diagonal matrix with scaling coefficients for each row, as follows:

$$\alpha = diag(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8).$$

### 4.2.2 Multi-objective optimization problem

The goal of resolving (4.18) is to find the best approximation based on multiple metrics, referred to as multi-objective or multi-criteria optimization

[101]. To guide the search for the best transform, we started with the problem outlined in [94, 32], which involves high deviation from orthogonality. By replacing $Y$ in (4.5) with (4.7), and the exact DTT matrix $T_N$ with another matrix $T$, we obtain:

$$X^* = T^T \cdot T \cdot X \cdot T^T \cdot T. \tag{4.19}$$

If the matrix $T$ is orthogonal, then $T^T = T^{-1}$ and $X^* = X$ because $T^T \cdot T = I$. The low deviation from orthogonality [30] ensures that the inverse matrix is very close to its transpose, meaning $T^T \approx T^{-1}$. This results in $X^*$ being approximately equal to the input $X$ ($X^* \approx X$) since $T^T \cdot T \approx I$. Consequently, the lowest deviation from orthogonality results in the lowest error possible.

To find the optimal low-complexity DTT approximation, we considered four criteria as figure-of-merits: (i) the deviations from orthogonality ($\delta$) [30], which is the most important criterion, (ii) the unified Cg [59], (iii) the transform efficiency ($\eta$) [59], and (iv) the MSE as a proximity measure. These metrics are crucial in assessing the transform's ability to preserve and compact energy, decorrelate data, and approach the exact transform [30]. The proposed methodology can be formulated as a multi-objective optimization problem:

$$\alpha^{OPT} = \text{ARG} \min_{0.5 \leq \alpha < 2.5} \{\delta(T^{approx}(\alpha)), MSE(T^{approx}(\alpha)), -C_g(T^{approx}(\alpha)), -\eta(T^{approx}(\alpha))\}, \tag{4.20}$$

with

$$T^{approx}(\alpha) = round(\alpha \cdot T^*), \tag{4.21}$$

$$\delta(A) = 1 - \frac{||diag(A \cdot A^T)||_F}{||A \cdot A^T||_F}, \tag{4.22}$$

where $T^*$ is the normalized matrix defined in (4.16), $||.||_F$ denotes the Frobenius norm for matrices [102], and Cg, $\eta$ and MSE are the coding gain, transform efficiency and mean square error defined in (4.49), (4.50) and (4.48), respectively.

The optimal low-complexity matrix can be obtained from the diagonal matrix of scaling parameters $\alpha^{OPT}$, as defined in (4.18). The criteria for maximization, Cg and $\eta$, are considered with a negative sign. To search for $\alpha^{OPT}$, linearly spaced values of $\alpha$ were considered in the interval $\frac{1}{2} \leq \alpha_i < \frac{5}{2}$, with a step size of 0.1. For each $\alpha$, a new matrix was generated using (4.21), and the metrics in (4.20) were calculated. The values of $\alpha$ for the matrices with the minimum metric values were saved in a set and ordered by the first and most important criteria $\delta$.

The almost perfect solution was achieved when $\alpha_1$ to $\alpha_8$ are within the intervals
$([\frac{1}{2}, \frac{3}{2}[, [\frac{3}{2}, \frac{21}{10}[, [\frac{3}{2}, \frac{21}{10}[, [\frac{7}{10}, \frac{7}{6}[, [\frac{13}{18}, \frac{13}{14}[, [\frac{23}{34}, \frac{23}{30}[, [\frac{3}{2}, \frac{5}{2}[, [\frac{1}{2}, \frac{5}{6}[)$, respectively. It is important to note that any $\alpha$ value within these intervals will result in the same approximation. For practical purposes, the selected optimal diagonal matrix of scaling parameters, $\alpha^{OPT}$, equals $diag(1, 2, 2, 1, \frac{3}{4}, \frac{3}{4}, 2, \frac{1}{2})$ . The proposed matrix $T_{p1}$ can be obtained directly using the following equation:

$$T_{p1}(\alpha^{OPT}) = round(D_1 \cdot T_0) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & -1 & 0 & 0 & 1 & 1 & 2 \\ 2 & 0 & -1 & -1 & -1 & -1 & 0 & 2 \\ -1 & 1 & 1 & 0 & 0 & -1 & -1 & 1 \\ 0 & -1 & 0 & 1 & 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 2 & -1 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \end{pmatrix}, \tag{4.23}$$

where $D_1 = \alpha^{OPT} \cdot D_0$.

### 4.2.3 Properties of the proposed DTT approximation

The properties of the proposed approximation can be summarized as follows:

- The inputs of the integer matrix $T_{p1}$ are in the set $P_2 = \{\pm 2, \pm 1, 0\}$, resulting in a low-complexity transformation $\hat{T}$ that requires only additions and bit-shifts, without the need for float multiplications. The scaling matrix will be incorporated into the quantization process.

- The matrix $T_{p1}$ has a small deviation from orthogonality, being near-orthogonal, as shown in (4.24) with $\delta < 1 - \frac{2}{\sqrt{5}}$ or $\delta < 0.1056$. This deviation from orthogonality, defined in (4.26), is small, which indicates that the transformation is nearly reversible and leads to efficient compression performance.

$$T_{p1} \cdot T_{p1}^T = \begin{pmatrix} 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 12 & 0 & -2 & 0 & -2 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 & 12 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix} \tag{4.24}$$

- The proposed approximation is near-orthonormal as shown in (4.25):

$$\hat{T} \cdot \hat{T}^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -0.29 & 0 & -0.17 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.29 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -0.17 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.25}$$

- The proposed transform has a low deviation from orthogonality, which means that it has a low computational complexity similar to the inverse transform. However, using the inverse ($\hat{T}^{-1}$) in the inverse 1D transform requires more hardware resources.

- The proposed approximation has the lowest error measured using the MSE value. Table 4.1 summarizes the comparison between the transformed domain using the transpose matrix and the inverse matrix in terms of the MSE.

- Due to its low deviation from orthogonality and low MSE value, the inverse of the proposed transform ($\hat{T}^{-1}$) is approximately equal to its transpose ($\hat{T}^T$). As a result, the inverse 1D transform can use the transpose matrix instead of the inverse.

Table 4.1: Comparison in terms of the deviation from orthogonality and the MSE between the transpose matrix and its exact inverse.

| Method | $\delta$ | $MSE(T^T,T^{-1})$ |
|---|---|---|
| DTT [86] | 0 | 0 |
| Signed Discret Cosine Transform (SDCT) [64] | 0.1056 | 0.0893 |
| $O_{15}$[94]. | 0.09 | 0.0385 |
| $O_{16}$[32]. | 0.024 | 0.0104 |
| $T_{p1}$ | **0.014** | **0.0026** |

## 4.3 Second proposition ($T_{p2}$) [103]

The first proposition [95] addressed the issue of orthogonality, where it offers the lowest deviation from orthogonality. It has an acceptable value of Cg,

rendering it suitable for image compression applications. However, it is important to note that the deviations of this approximation are located in the low frequency elements. Consequently, it is critical to carefully consider the deviation from orthogonality while selecting DTT approximations for image compression applications.

### 4.3.1 Proposed modified deviation-from-orthogonality ($m\delta$)

In transform-based compression, quantization is typically the only step that causes loss. However, when using a quasi-orthogonal kernel transform, loss can also result from employing the transpose matrix. To minimize these losses, it is crucial to reduce deviations from orthogonality. The deviation from orthogonality metric ($\delta$), defined in equation (4.23), is generally used in this context

$$\delta(A) = 1 - \frac{||diag(A)||_F}{||A||_F}, \tag{4.26}$$

where $||.||_F$ denotes the Frobenius norm for matrices [102].

The deviations from orthogonality of a transform approximation can occur in different areas, corresponding to low or high frequencies. The critical area for image and video compression is the low frequency region as it contains the most important information that needs to be preserved. In such a region, these deviations can negatively impact the compression efficiency of the algorithm. However, $\delta$ does not take into consideration the location of deviations, which is very important. Therefore, we introduce a modified deviation from orthogonality ($m\delta$) to emphasize deviations in the crucial area where the base frequencies are located. This is achieved by adding a new parameter $\sigma$, defined as follows:

$$\sigma_{i,j} = \frac{1}{min(i,j)}, i, j = 1, \ldots, N, \tag{4.27}$$

where $\sigma$ is a matrix of size $N \times N$, $N$ is the size of the matrix $A$ in Equation (4.26), and $min$ returns the minimum of two input values. $m\delta$ is then defined as follows:

$$m\delta(A) = 1 - \frac{||diag(A \odot \sigma)||_F}{||A \odot \sigma||_F}, \tag{4.28}$$

where $\odot$ is the elements-wise multiplication.

To better illustrate the proposed $m\delta$, let us consider an example. We will use the matrix $A_1$ shown in (4.29), which corresponds to the transform kernel in (4.14). Additionally, we introduce a modified matrix $A_2$ in (4.30),

where some of the deviations were shifted from the low-frequency region to the high-frequency area. The shifted elements are highlighted in matrix $A_2$.

$$A_1 = C_{16} \cdot C_{16}^{\top} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \mathbf{0.13} & 0 & \mathbf{0.15} & 0 & \mathbf{0.18} \\ 0 & 0 & 1 & 0 & \mathbf{0.17} & 0 & \mathbf{-0.17} & 0 \\ 0 & \mathbf{0.13} & 0 & 1 & 0 & \mathbf{0.12} & 0 & 0 \\ 0 & 0 & \mathbf{0.17} & 0 & 1 & 0 & \mathbf{0.17} & 0 \\ 0 & \mathbf{0.15} & 0 & \mathbf{0.12} & 0 & 1 & 0 & \mathbf{-0.17} \\ 0 & 0 & \mathbf{-0.17} & 0 & \mathbf{0.17} & 0 & 1 & 0 \\ 0 & \mathbf{0.18} & 0 & 0 & 0 & \mathbf{-0.17} & 0 & 1 \end{pmatrix} \tag{4.29}$$

$$A_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \underline{\mathbf{0}} & 0 & \mathbf{0.15} & 0 & \mathbf{0.18} \\ 0 & 0 & 1 & 0 & \mathbf{0.17} & 0 & \mathbf{-0.17} & 0 \\ 0 & \underline{\mathbf{0}} & 0 & 1 & 0 & \mathbf{0.12} & 0 & 0 \\ 0 & 0 & \mathbf{0.17} & 0 & 1 & 0 & \mathbf{0.17} & 0 \\ 0 & \mathbf{0.15} & 0 & \mathbf{0.12} & 0 & 1 & 0 & \mathbf{-0.17} \\ 0 & 0 & \mathbf{-0.17} & 0 & \mathbf{0.17} & 0 & 1 & \underline{\mathbf{0.13}} \\ 0 & \mathbf{0.18} & 0 & 0 & 0 & \mathbf{-0.17} & \underline{\mathbf{0.13}} & 1 \end{pmatrix} \tag{4.30}$$

Table 4.2 demonstrates that the proposed $m\delta$ provides a more accurate representation of the impact of deviations on a given non orthogonal transformation. As a result, it is a valuable tool for evaluating and improving the performance of quasi-orthogonal kernels in transform-based compression applications.

Table 4.2: Comparison of deviation from orthogonality.

| Method | $\delta$ | $m\delta$ |
|---|---|---|
| $A_1(O_{16})$[32] | 0.0241 | 0.0175 |
| $A_2$ | 0.0241 | **0.0151** |

### 4.3.2 Multi-objective optimization problem

This study employs the scale-and-round approach explained in Section 4.2.1 and a multi-objective optimization algorithm to minimize deviations in the DTT matrix. The optimization process follows the following rules:

1. Transfer deviations to the high-frequency elements. As rows and columns in $T \cdot T^{\top}$ are related, deviations should be moved from row and column 1 to row and column $N$.

2. Optimize coding performance and proximity to the exact DTT.

Adhering to these rules, the best approximation is sought using a multi-objective optimization algorithm based on the following criteria: (i) the $m\delta$, the most important factor to satisfy rule 1; (ii) the Cg) [30]; and (iii) the MSE as a proximity measure to fulfill rule 2.

These figures of merit are significant because they ensure low deviation from orthogonality, the transform's ability to preserve and compact energy, data decorrelation, and proximity to the exact transform. This methodology can be formulated as the following multi-objective optimization problem: "Minimize $m\delta$, Cg, and MSE subject to constraints on the DTT matrix elements.", which can be formulated by the following equation:

$$\begin{aligned} \alpha^{O} = \text{ARG} \min_{\alpha_{min} \leq \alpha \leq \alpha_{max}} \{ & m\delta(T^{*}(\alpha) \cdot T^{*\top}(\alpha)), \\ & MSE(T^{*}(\alpha)), -C_g(T^{*}(\alpha))\}, \end{aligned} \tag{4.31}$$

where $m\delta$ is defined in (4.28) and $\alpha^{O}$ the optimal diagonal matrix of scaling parameters is denoted by $\alpha^{O}$ and is given by Equation (4.18). Since Cg is to be maximized, we consider it with a negative sign.

To ensure that the values of $T_{p1}$ are in the set $P_2 = \{\pm 2, \pm 1, 0\}$, the maximum value of each row of $T_{p1}$ should not exceed 2.5 (as $round(2.5) = 3$), we set the upper bound for $\alpha$ of row 3 and 5 as follows:

$$\alpha_{max} = \frac{2.5}{max_r(abs(T_8(\{2,4\})))} \approx \{4.6, 4.7\}$$

Note that the maximum value of the first row should be ignored, as all its values are equal and only represent an up-scaling factor. Additionally, each row of $T^{*}(\alpha)$ must not be null, meaning that the maximum value of each row ($max_r$) should be at least 0.5, as expressed in the following equation:

$$\alpha_{min} = \frac{0.5}{max_r(abs(T_8(\{2,4\})))} \approx \{0.9, 0.9\}$$

In the optimization process, we use a step size of $\alpha_{step} = 0.1$. Doing so, the optimal values of $\alpha$ belong to the following intervals:

$$(\alpha_2^{O}, \alpha_4^{O}) \in \left( \left[ \frac{\sqrt{42}}{5}, \frac{\sqrt{42}}{3} \right[ , \left[ \frac{3\sqrt{154}}{13}, \frac{\sqrt{154}}{3} \right] \right)$$

It is important to note that any value of $\alpha^O$ within the specified intervals would yield the same approximation. For practical purposes, we have chosen $\alpha^O = diag(1, 1, 2, 1, 3, 1, 1, 1)$. With this value, the proposed matrix $T_{p1}$ can be directly obtained using (4.18). The resulting matrix is presented in the following equation:

$$T_{p2}(\alpha^O) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & -1 & 0 & 0 & 1 & 1 & 2 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ -1 & 1 & 1 & 0 & 0 & -1 & -1 & 1 \\ 1 & -2 & 0 & 1 & 1 & 0 & -2 & 1 \\ 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 2 & -1 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \end{pmatrix} \tag{4.32}$$

### 4.3.3 Properties of the proposed transformations

The proposed transformation $T_{p2}$ has several important properties, as described below.

1. Complexity: Since the inputs of the integer matrix $T_{p2}$ are in the set $P_2 = \{\pm 2, \pm 1, 0\}$, the proposed transformation inherits the same complexity as $T_{p2}$. This means that $T_{p2}$ does not require any floating-point multiplications and only additions and shifts are needed.

2. Near-Orthogonality: The matrix $T_{p2}$ has a small deviation from orthogonality and is considered near-orthogonal as $m\delta(T_{p2}) < m\delta(SDCT)$ [64], according to [50, 79, 94, 32]. Table 4.3 summarizes the deviations of different transforms and shows that the proposed approximation has the lowest deviation compared to the other transforms. In this study, the correlation factor $\rho$ is fixed at 0.95, a value established as a dependable approximation for natural images in previous literature [59].

3. Low Complexity Inverse: Since the proposed transformation matrix $T_{p2}$ is near-orthogonal and has a small deviation from orthogonality, the complexity of the inverse kernel matrix is almost equal to the arithmetic complexity of the matrix itself. This means that the inverse kernel matrix can be used.

4. Use of Transpose: The small deviations from orthogonality of the proposed transformation make the inverse of the matrix approximately equal to its transpose (as shown in Table 4.3). This means that when

Table 4.3: Comparison of modified deviation from orthogonality and MSE between the transpose kernel and the inverse (with $\rho = 0.95$).

| Method | $\delta$ | $m\delta$ | $MSE(T^{\top}, T^{-1})$ |
|---|---|---|---|
| DTT[86] | 0 | 0 | 0 |
| $SDCT$[64] | 0.1056 | 0.0845 | 0.0893 |
| $O_{15}$[94] | 0.09 | 0.0541 | 0.0385 |
| $O_{16}$[32] | 0.024 | 0.0175 | 0.0104 |
| $T_{p1}$[95] | **0.014** | 0.008 | **0.0026** |
| $T_{p2}$ | **0.014** | **0.0067** | **0.0026** |

the transformation uses the transpose matrix, the error of the transformation becomes small. Therefore, the transpose matrix of the kernel can be used instead of the inverse, resulting in good compression efficiency for a matrix with an acceptable Cg.

## 4.4 2D Transformations

Polar decomposition can be used to derive an orthonormal or quasi-orthonormal matrix $T$ from a low-complexity matrix. The goal of this process is twofold: first, to ensure that the matrix is reversible when it is orthogonal; and second, to minimize errors when the transpose is used as an approximation to the inverse transform matrix. When the matrix is orthonormal, the 1D forward and inverse transforms require the same complexity. However, when the matrix is not orthonormal, the complexity is almost the same for both the forward and inverse transforms. The 1D forward transforms $T_i$ associated with the matrices $T_{pi}$ can be expressed as follow:

$$T_i = S_i \cdot T_{pi}, \tag{4.33}$$

where $S_i = \sqrt{(T_{pi} \cdot T_{pi}^{\top})^{-1}}$ is the scaling matrix used in the orthogonalization procedure [30] and $i = \{1, 2\}$.

Both approximations $T_{p1}$ and $T_{p2}$ are near-orthogonal. It is necessary to approximate $S_i$ by removing the off-diagonal items. The resulting matrices ($\hat{S}_i$) are given by (4.34) and (4.35):

$$\begin{aligned} \hat{S}_1 &= \sqrt{diag(T_{p1} \cdot T_{p1}^{\top})^{-1}} \\ &= diag(\frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{3}}, \frac{1}{2\sqrt{3}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2\sqrt{3}}, \frac{1}{\sqrt{2}}) \end{aligned} \tag{4.34}$$

$$\begin{aligned}\hat{S}_2 &= \sqrt{diag(T_{p2} \cdot T_{p2}^{\top})^{-1}} \\ &= diag(\frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{3}}, \frac{1}{2}, \frac{1}{\sqrt{6}}, \frac{1}{2\sqrt{3}}, \frac{1}{2}, \frac{1}{2\sqrt{3}}, \frac{1}{\sqrt{2}})\end{aligned} \tag{4.35}$$

This process ensures that the scaling factor $\hat{S}_i$ can be adjusted during the quantization step.

As shown in Property 4 in Subsection 4.3.3, the proposed matrix is nearly orthogonal. This property offers two options for the 2D transform, as described in Equation (4.5). The first option is to use $T_{p2}^{\top}$ as the inverse transform, resulting in a lower complexity encoder. With this approach, the 1D forward and inverse transforms use the same matrix, leading to a single architecture. Alternatively, the second option is to use $T_{p2}^{-1}$ directly as the inverse transform. This approach can improve quality, but at the cost of increased encoder complexity. With this option, two separate architectures are required. In this chapter, we present and evaluate both transformations.

The first and the second proposed transformations, denoted as $P_{T1}$ and $P_{T2}$, uses the transformation kernel as defined in Equation (4.36) for both the forward and inverse 1D transforms.

$$T_i = \hat{S}_i \cdot T_{pi} \tag{4.36}$$

Using this transformation kernel, the output $Y$ of the 2D transform is calculated as follows:

$$Y = T_i \cdot X \cdot T_i^{\top} = \hat{S}_i \cdot T_{pi} \cdot X \cdot (\hat{S}_i \cdot T_{pi})^{\top} = \hat{S}_i \cdot (T_{pi} \cdot X \cdot T_{pi}^{\top}) \cdot \hat{S}_i, \tag{4.37}$$

where $X$ represents the input data of size $N \times N$, $i = \{1, 2\}$ and $\hat{S}$ is the approximate diagonal matrix defined in (4.35).

The third proposed transformation, denoted as $P_{T3}$, utilizes $T_{p2}$ in the forward 1D transform and $T_{p2}^{-1}$ in the inverse 1D transform. However, since $T_{p2}^{-1}$ has real elements, it cannot be implemented as is. To overcome this issue, we propose the following decomposition:

$$T_{p2}^{-1} = T_{p3} \cdot D \tag{4.38}$$

$$T_{p3} = \begin{pmatrix} 1 & -2 & 9 & -1 & 3 & 0 & -1 & 0 \\ 1 & -1 & 1 & 1 & -5 & 1 & -1 & 0 \\ 1 & -1 & -3 & 1 & -1 & -1 & 3 & 0 \\ 1 & 0 & -7 & 0 & 3 & 0 & -1 & 1 \\ 1 & 0 & -7 & 0 & 3 & 0 & -1 & -1 \\ 1 & 1 & -3 & -1 & -1 & 1 & 3 & 0 \\ 1 & 1 & 1 & -1 & -5 & -1 & -1 & 0 \\ 1 & 2 & 9 & 1 & 3 & 0 & -1 & 0 \end{pmatrix} \tag{4.39}$$

$$D_3 = \frac{1}{2} \cdot diag(\frac{1}{4}, \frac{1}{6}, \frac{1}{16}, \frac{1}{3}, \frac{1}{16}, \frac{1}{2}, \frac{1}{8}, 1), \tag{4.40}$$

where $T_{p3}$ is an integer matrix, which makes the computation of $P_{T3}$ simpler and more efficient, and $D$ is a diagonal matrix that can be shifted in the quantization step. Using this decomposition, the output $Y$ of the 2D forward transform is calculated as follows:

$$Y = \hat{S}_2 \cdot T_{p2} \cdot X \cdot (\hat{S}_2 \cdot T_{p2})^{-1} = \hat{S}_2 \cdot T_{p2} \cdot X \cdot T_{p3} \cdot D_3 \cdot \hat{S}_2^{-1} = T_{p2} \cdot X \cdot T_{p3} \cdot \hat{S}_3, \tag{4.41}$$

with:

$$\hat{S}_3 = D_3 \cdot \hat{S}_2^{-1} = \frac{1}{2} \cdot diag(\frac{\sqrt{2}}{2}, \frac{\sqrt{3}}{3}, \frac{1}{8}, \frac{\sqrt{6}}{3}, \frac{\sqrt{3}}{8}, 1, \frac{\sqrt{3}}{4}, \sqrt{2}) \tag{4.42}$$

However, in several contexts, such as JPEG-based image compression applications [86, 29, 30], diagonal matrices like $\hat{S}_i$ and $D$ do not significantly contribute to the computational cost of a transformation and can be embedded in the quantization step.

Therefore, equations (4.37) and (4.41) can be described as follows:

$$P_{T1} \longrightarrow Y = \hat{S}_1 \cdot T_{p1} \cdot X \cdot T_{p1}^\top \cdot \hat{S}_1 = T_{p1} \cdot X \cdot T_{p1}^\top \odot (s_1 \cdot s_1^\top) \tag{4.43}$$

$$P_{T2} \longrightarrow Y = \hat{S}_2 \cdot T_{p2} \cdot X \cdot T_{p2}^\top \cdot \hat{S}_2 = T_{p2} \cdot X \cdot T_{p2}^\top \odot (s_2 \cdot s_2^\top) \tag{4.44}$$

$$P_{T3} \longrightarrow Y = \hat{S}_2 \cdot T_{p2} \cdot X \cdot T_{p3} \cdot \hat{S}_3 = T_{p2} \cdot X \cdot T_{p3} \odot (s_2 \cdot s_3^\top), \tag{4.45}$$

where $s_1$, $s_2$ and $s_3$ are column vectors containing the diagonal elements of the scaling matrices $\hat{S}_1$, $\hat{S}_2$ and $\hat{S}_3$, respectively. The symbol $\odot$ denotes element-wise multiplication. Based on Equations (4.43), (4.44) and (4.45), $(s_i \cdot s_j^\top)$ can easily be integrated into the computation of the quantization.

## 4.5 Performance assessment

### 4.5.1 Proposed fast algorithms and arithmetic complexities

This section addresses the efficient computation of the proposed approximation, $T_{p1}$. Direct computation of the approximation from (4.43) requires 34 additions and 6 shifts operations, making it too complex. To resolve this, a sparse matrix factorization is employed to reduce the number of operations. The proposed approximation can then be calculated by multiplying sparse matrices based on the usual butterfly structures [63] :

$$T_{p1} = P \cdot A_2 \cdot A_1 \cdot B, \tag{4.46}$$

where

$$B = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \quad A_1 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 2 \end{pmatrix} \quad P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$T_{p2} = \hat{P} \cdot \hat{A}_2 \cdot \hat{A}_1 \cdot \hat{B}, \tag{4.47}$$

with:

$$\hat{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \qquad \hat{A}_1 = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\hat{A}_2 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \qquad \hat{P} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Table 4.4: Fast algorithm of the proposed 1D transform $T_{p1}$.

| Step 1 | Step 2 | Step 3 |
|---|---|---|
| | | $y_1 = u_1 + u_2 + u_{t1};$ |
| $u_1 = x_1 + x_8;$ | $v_1 = x_1 - x_8;$ | $y_2 = (-v_1 \ll 1) - u_{t2};$ |
| $u_2 = x_2 + x_7;$ | $v_2 = x_2 - x_7;$ | $y_3 = u_1 \ll 1 - u_{t1};$ |
| $u_3 = x_3 + x_6;$ | $v_3 = x_3 - x_6;$ | $y_4 = u_{t2} - v_1;$ |
| $u_4 = x_4 + x_5;$ | $v_4 = x_4 - x_5;$ | $y_5 = u_4 - u_2;$ |
| $u_{t1} = u_3 + u_4;$ | $u_{t2} = v_2 + v_3;$ | $y_6 = v_2 - v_3;$ |
| | | $y_7 = u_3 \ll 1 - u_2 - u_4;$ |
| | | $y_8 = v_4;$ |

Table 4.4 and Figure 4.1 show the fast algorithm and the SFG corresponding to the above matrix decomposition of the proposed matrix $T_{p1}$, respectively. Table 4.5 and Figure 4.2 show the fast algorithm and the SFG corresponding to the above matrix decomposition of the proposed matrix $T_{p2}$, respectively. Table 4.6 present the fast algorithm of the matrix $T_{p3}$. Table 4.7 assesses the arithmetic complexity and gives a comparison in terms of a number of addition and bit-shift operations. Note that $X = [x_1, x_2, \ldots, x_8]$ and $Y = [y_1, y_2, \ldots, y_8]$ are the 8-point input and output vectors, respectively.

Table 4.7 compares the arithmetic complexity of the proposed transform with other existing techniques in terms of the number of additions and bit-shift operations. The results indicate that the complexity of the forward

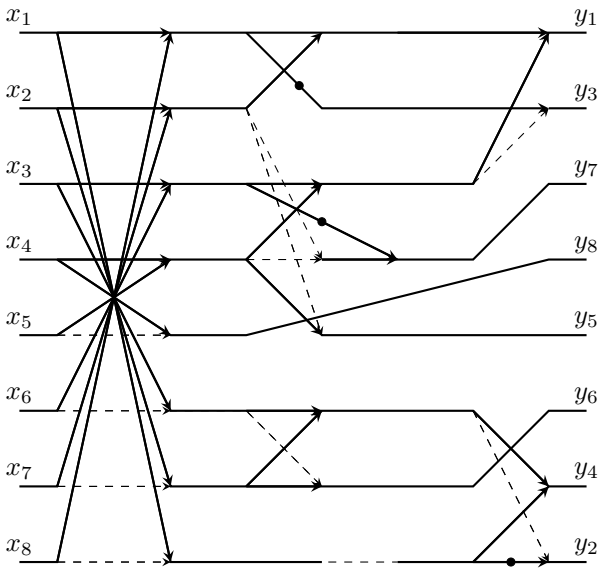



Figure 4.1: SFG for $T_{p1}$. Input data $x_n$, $n = 1, 2, \ldots, 8$, output $y_m$, $m = 1, 2, \ldots, 8$. Dashed lines and black nodes represent multiplications by -1 and 2 respectively.

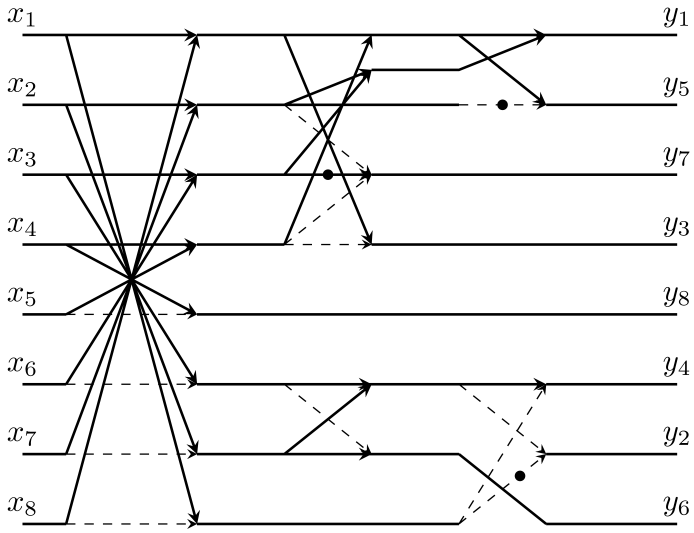



Figure 4.2: SFG for $T_{p2}$. Input data $x_n$, $n = 1, 2, \ldots, 8$, relates to output $y_m$, $m = 1, 2, \ldots, 8$. Dashed lines and black nodes represent multiplications by -1 and 2 respectively.

Table 4.5: Fast algorithm for the proposed $T_{p2}$ matrix

| Step 1 | Step 2 | Step 3 |
|---|---|---|
| | | $y_1 = u_2 + u_3 + t_1;$ |
| $u_1 = x_1 + x_8;$ | $v_1 = x_1 - x_8;$ | $y_2 = (-v_1 \ll 1) - t_2;$ |
| $u_2 = x_2 + x_7;$ | $v_2 = x_2 - x_7;$ | $y_3 = u_1 - u_4;$ |
| $u_3 = x_3 + x_6;$ | $v_3 = x_3 - x_6;$ | $y_4 = t_2 - v_1;$ |
| $u_4 = x_4 + x_5;$ | $v_4 = x_4 - x_5;$ | $y_5 = t_1 - (u_2 \ll 1);$ |
| $t_1 = u_1 + u_4;$ | $t_2 = v_3 + v_2;$ | $y_6 = v_2 - v_3;$ |
| | | $y_7 = (u_3 \ll 1) - u_2 - u_4;$ |
| | | $y_8 = v_4;$ |

Table 4.6: Fast algorithm for the proposed 1D inverse transform $T_{p3}$

| Step 1 | Step 2 | Step 3 |
|---|---|---|
| $u_1 = x_1 + x_8;$ | | $y_1 = t_5;$ |
| $u_2 = x_2 + x_7;$ | $t_1 = u_1 + u_4$ | $y_2 = (-v_1 \ll 1) - t_4;$ |
| $u_3 = x_3 + x_6;$ | $t_2 = u_2 + u_3;$ | $y_3 = (t_3 \ll 3) - t_6;$ |
| $u_4 = x_4 + x_5;$ | $t_3 = u_1 - u_4;$ | $y_4 = t_4 - v_1;$ |
| $v_1 = x_1 - x_8;$ | $t_4 = v_2 + v_3;$ | $y_5 = (t_7 \ll 2) - t_5;$ |
| $v_2 = x_2 - x_7;$ | $t_5 = t_1 + t_2;$ | $y_6 = v_2 - v_3;$ |
| $v_3 = x_3 - x_6;$ | $t_6 = (u_3 \ll 2) - t_5;$ | $y_7 = t_6;$ |
| $v_4 = x_4 - x_5;$ | $t_7 = t_1 - u_2$ | $y_8 = v_4;$ |

transform of the proposed approximation is comparable to that of the one presented in [94]. However, the proposed approximation has a lower complexity in the inverse operation. When compared to the DTT approximations [94], [32], as well as to the exact DTT [86], it can be seen that the number of operations has been reduced by 22%, 26.67%, and 69.86%, respectively. While some elements of the $T_{p1}$ matrix can be adjusted to reduce the computational complexity further, this would negatively affect compression efficiency. Therefore, the additional computational cost has been kept in order to maintain the best compression results, as demonstrated in the final section.

### 4.5.2 MSE, Cg and transform efficiency

The proposed approximation has been evaluated in comparison to previous DTT approximations [94, 32] and the exact DTT [86] in terms of both similarity and coding performance. The MSE [59], defined in (4.48), has been used to measure similarity. Meanwhile, the coding performance has been assessed using the coding gain Cg [59] and the transform efficiency ($\eta$) [59,

Table 4.7: Arithmetic complexity comparison for 8-point DTT approximations (1D and 2D).

| | 8-point 1D | | | 2D | | |
|---|---|---|---|---|---|---|
| Approximation | adds | b-shifts | Total | adds | b-shifts | Total |
| DTT[86] Forward | 44 | 29 | 73 | 352 | 232 | 584 |
| DTT[86] Inverse | 44 | 29 | 73 | 352 | 232 | 584 |
| **Total** | | | **146** | | | **1168** |
| $O_{15}$[94]. Forward | 20 | 0 | 20 | 160 | 0 | 160 |
| $O_{15}$[94]. Inverse | 29 | 8 | 37 | 232 | 64 | 296 |
| **Total** | | | **57** | | | **456** |
| $O_{16}$[32]. Forward | 24 | 6 | 30 | 192 | 48 | 240 |
| $O_{16}$[32]. Inverse | 24 | 6 | 30 | 192 | 48 | 240 |
| **Total** | | | **60** | | | **480** |
| $P_{T1}$ Forward ($T_{p1}$) | 19 | 3 | 22 | 152 | 24 | 176 |
| $P_{T1}$ Inverse ($T_{p1}$) | 19 | 3 | 22 | 152 | 24 | 176 |
| **Total** | | | **44** | | | **352** |
| $P_{T2}$ Forward ($T_{p2}$) | 19 | 3 | 22 | 152 | 24 | 176 |
| $P_{T2}$ Inverse ($T_{p2}$) | 19 | 3 | 22 | 152 | 24 | 176 |
| **Total** | | | **44** | | | **352** |
| $P_{T3}$ Forward ($T_{p2}$) | 19 | 3 | 22 | 152 | 24 | 176 |
| $P_{T3}$ Inverse ($T_{p3}$) | 20 | 4 | 24 | 160 | 32 | 192 |
| **Total** | | | **46** | | | **368** |

104], as defined in Eqs. (4.49) and (4.50), respectively.

Let $C = S \cdot T$ and $R_y = C \cdot R_x \cdot C^T$, where $R_x$ is the covariance of $x$ whose elements are given by $\rho^{|i-j|}$, $i, j = 1, 2, \ldots, 8$ and a correlation factor $\rho = 0.95$.

$$MSE(C_1, C_2) = \frac{1}{N} Trace\{(C_1 - C_2) \cdot R_x \cdot (C_1 - C_2)^T\}, \tag{4.48}$$

where Trace denotes the trace of a matrix, which is defined as the sum of its diagonal elements.

$$C_g = 10log_{10} \frac{\frac{1}{N}\sum_{i=0}^{N-1} \sigma_{x_i}^2}{\left(\prod_{i=0}^{N-1} \sigma_{x_i}^2 ||f_i||^2\right)^{\frac{1}{N}}}, \tag{4.49}$$

where $N$ is the number of transform coefficients, $\sigma_{x_i}^2$ is the variance of $i^{th}$ transform coefficient being the $i^{th}$ diagonal element of the matrix $R_y$ and

$||f_i||$ is the 2-norm of the $i^{th}$ basis function of the transform matrix.

$$\eta = \frac{\sum_{i=1}^{N} |r_{i,i}|}{\sum_{i=1}^{N} \sum_{j=1}^{N} |r_{i,j}|} \cdot 100, \tag{4.50}$$

where $r_{i,j}$ are elements of $R_y$.

As demonstrated in Table 5.1, the proposed approximations has a competitive coding gain Cg and transform efficiency $\eta$. In terms of coding gain, they outperforms the DTT approximation in [94] by 1.25 dB, 0.33 dB and 1.63 dB compared to $P_{t1}$, $P_{t3}$ and $P_{t2}$ . The calculated MSE indicates that the proposed approximations are close to the exact DTT and are a better approximation than the DTT approximation in [94]. Additionally, it still maintains a low computational complexity, as previously discussed in Subsection 4.5.1.

Table 4.8: Performance assessment.

| Method | Cg[$db$] | $\eta$ | MSE |
|---|---|---|---|
| DTT[86] | 8.68 | 92.86 | - |
| WHT | 7.95 | - | – |
| Haar | 7.94 | - | – |
| SDCT[64] | 7.79 | - | – |
| $O_{15}$[94] | 6.6 | 83.50 | 0.0149 |
| $O_{16}$[32] | 8.57 | 89.52 | 0.0022 |
| $P_{T1}$ | 7.85 | 85.77 | 0.0115 |
| $P_{T2}$ | 8.23 | 89.02 | 0.008 |
| $P_{T3}$ | 6.93 | 91.73 | 0.008 |

On the other hand, as shown in Table 5.1, the approximation in [32] outperforms the proposed ones in terms of coding gain and transform efficiency. However, it has a higher deviation from orthogonality compared to the proposed approximations, as indicated in Table 4.3. This higher deviation results in a difference between the inverse and the transpose, affecting the image quality during the reconstruction process. In contrast, the proposed approximations have a lower coding gain and efficiency, but they have the advantage of a lower deviation from orthogonality, leading to a smaller error between the inverse and the transpose. Additionally, the proposed approximations have a lower arithmetic complexity, as shown in Table 4.7.

### 4.5.3 Hardware implementation

According to Table 4.7, the proposed DTTs have a lower arithmetic complexity. To verify this advantage, the proposed DTTs and the 1-D DTT approximation[32] were implemented on the Xilinx Virtex-6 XC6VSX475T-1FF1759-2 using Xilinx ISE Design Suite 12.2. The results displayed in Table 5.3, compare the use of slice registers, slice Look-Up Table (LUT)s, and Flip Flops, as the Xilinx tool does not report the number of Configurable Logic Block (CLB)s. These results show that the proposed approximation uses fewer resources and consumes less power.

Table 4.7 demonstrates that the proposed DTTs significantly reduces the arithmetic complexity when compared to [32]. To investigate the practical implications of this advantage, we implemented the 1-D 8-points transformation process of the proposed DTT approximations ($T_{p1}$ and $T_{p2}$ presented by the algorithms in Tables 4.5 and 4.6, respectively), as well as $O_{16}$ [32], on the Xilinx Virtex-6 XC6VSX475T-1FF1759-2 using Xilinx ISE Design Suite 12.2. Table 5.3 reports the results of our experiments, comparing the slice registers, slice LUTs, Flip Flops (FF), Tcpd (ns), Fmax (MHz), power (W), and power/MHz (mW/MHz) for each implementation.

Table 4.9: Hardware resource consumption for 1D forward transformation using Xilinx Virtex-6 XC6VSX475T-1FF1759

| | $O_{16}$ [32] | $T_{p1}$ [95] | $T_{p2}$[103] |
|---|---|---|---|
| Slice Registers | 212 | 188 | 179 |
| Slice LUTs | 248 | 177 | 173 |
| FF | 316 | 262 | 253 |
| Tcpd (ns) | 1.924 | 1.908 | 1.913 |
| Fmax (MHz) | 519.656 | 524.054 | 522.657 |
| Power (W) | 4.686 | 4.63 | 4.633 |
| Power/MHz (mW/MHz) | 9.017 | 8.835 | 8.86 |

Our experiments demonstrate that the proposed DTT approximation $T_{p2}$ surpasses $O_{16}$ in hardware resources, delay (Tcpd), and power consumption. The delay is reduced by 11 ps, and there is a power consumption gain of 1.74% per MHz. While the absolute difference in power consumption between the proposed approximations and $O_{16}$ (53 $mW$ for $T_{p2}$ and 56 $mW$ for $T_{p1}$) may seem insignificant for the 1-D 8-point transform, it should be noted that this value may increase significantly for the 2-D transform, reaching up to 848 mW, which could lead to hundreds of watts for the smallest images.

When comparing the DTT approximation $T_{p2}$ with $T_{p1}$, we observe that $T_{p2}$ requires fewer hardware resources, while the delay and power consump-

tion are only slightly similar. Our findings indicate a marginal loss of 0.26% in delay (Tcpd) and 0.28% in power consumption per MHz for $T_{p2}$ compared to $T_{p1}$. Overall, our experiments confirm that the proposed DTT approximations are a competitive alternative to existing approximations, with considerable advantages in terms of reduced arithmetic complexity and lower power consumption per MHz. These findings have important implications for signal processing applications, particularly in image and video compression, where efficient coding and decorrelation properties are essential for real-time applications and embedded systems.

## 4.6 Applications in image compression

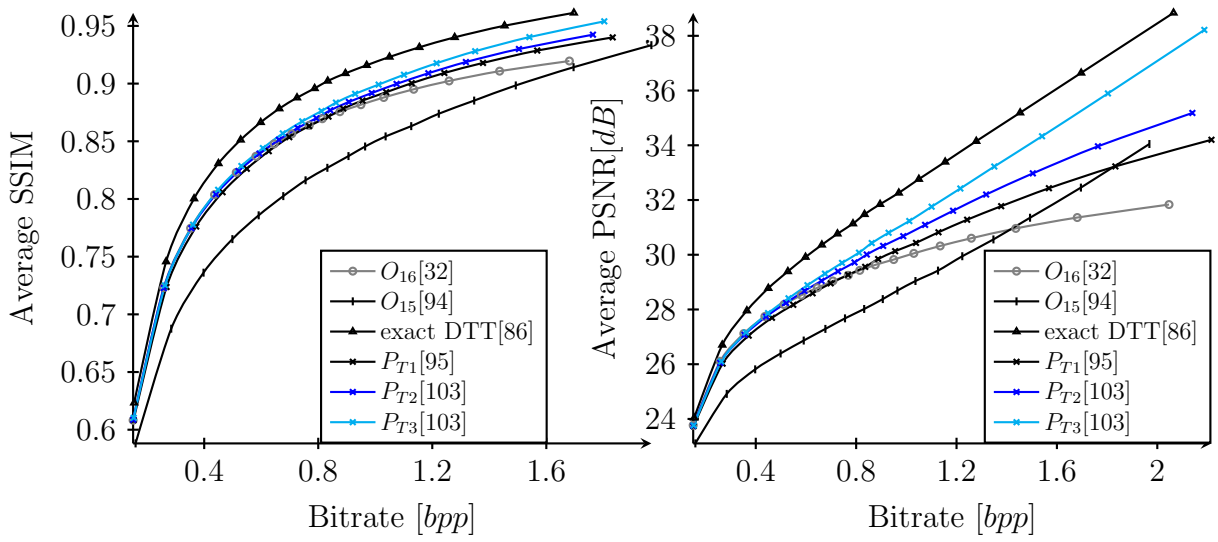
To further evaluate the effectiveness of the proposed algorithms, we conducted a JPEG compression experiment using a dataset of 47 8-bit images obtained from a public image bank [66]. The images were partitioned into $8 \times 8$ blocks and transformed using the proposed matrices and matrices from [86, 94, 32, 95]. The transformed images were then compressed using a range of quality factors, and their quality was assessed using two image quality metrics, namely PSNR and SSIM. These metrics were calculated for each quality factor ($QF$) value, ranging from 4 to 88 with a step of 4, for all 47 images. Subsequently, the average image quality metrics and bit-rate were computed based on these calculations.

Fig. 4.3 demonstrate that the proposed DTT approximations outperform existing methods in terms of PSNR and SSIM for bit-rates greater than 0.5 bpp. Notably, the proposed $P_{T3}$ approximation achieves the highest PSNR quality among all the approximations for all bit-rates, closely approaching the PSNR quality of the exact DTT. For bit-rates around 1.6 bpp, $P_{T3}$ exhibits considerable PSNR gains, up to more than 4 dB, 3 dB, and 2.5 dB, respectively, compared to $O_{16}$ [32], $P_{T1}$ [95] and $O_{15}$ [94]. Additionally, the proposed $P_{T2}$ and $P_{T3}$ approximations outperform $O_{15}$ [94] for low bit-rates up to 0.5 bpp. The results emphasize the suitability of the proposed approximations for applications in remote sensing and embedded systems with limited resources.

Moreover, the complexity analysis reveals that $P_{T2}$ exhibits the lowest complexity among all the transformations since it employs the same encoder architecture, while $O_{15}$ employs two distinct transformations. However, $P_{T2}$ exhibits PSNR degradation at higher bit-rates compared to $P_{T3}$ and $O_{15}$ [94]. Nonetheless, we emphasize that the proposed approximations are efficient and demonstrate superior performance, making them a viable alternative for image compression applications.

Fig. 4.3b shows that the proposed approximations outperform $O_{16}$ [32] and $O_{15}$ [94], except for bit-rates greater than 1.5 bpp where degradation is noticeable for $P_{T2}$ compared to $O_{15}$. Notably, the proposed approximation $P_{T3}$ achieves the highest PSNR quality among all the approximations for all bit-rates, closely approaching the PSNR quality of the exact DTT [86].

Furthermore, Fig. 4.3b reveals that for low bit-rates up to 0.5 bpp, the proposed $P_{T1}$, $P_{T2}$ and $P_{T3}$ approximations outperform $O_{15}$ [94], while $P_{T2}$ demonstrates performance similar to $O_{16}$ [32]. It is worth noting that bit-rates below 0.5 bpp are particularly suitable for embedded systems. This highlights the suitability of the proposed approximations for embedded systems with limited resources such as [39]. Despite the PSNR degradation of $P_{T1}$ and $P_{T2}$ at higher bit-rates, it should be emphasized that it exhibits the lowest complexity among all the transformations since it employs the same encoder architecture, while $O_{15}$ employs two distinct transformations.



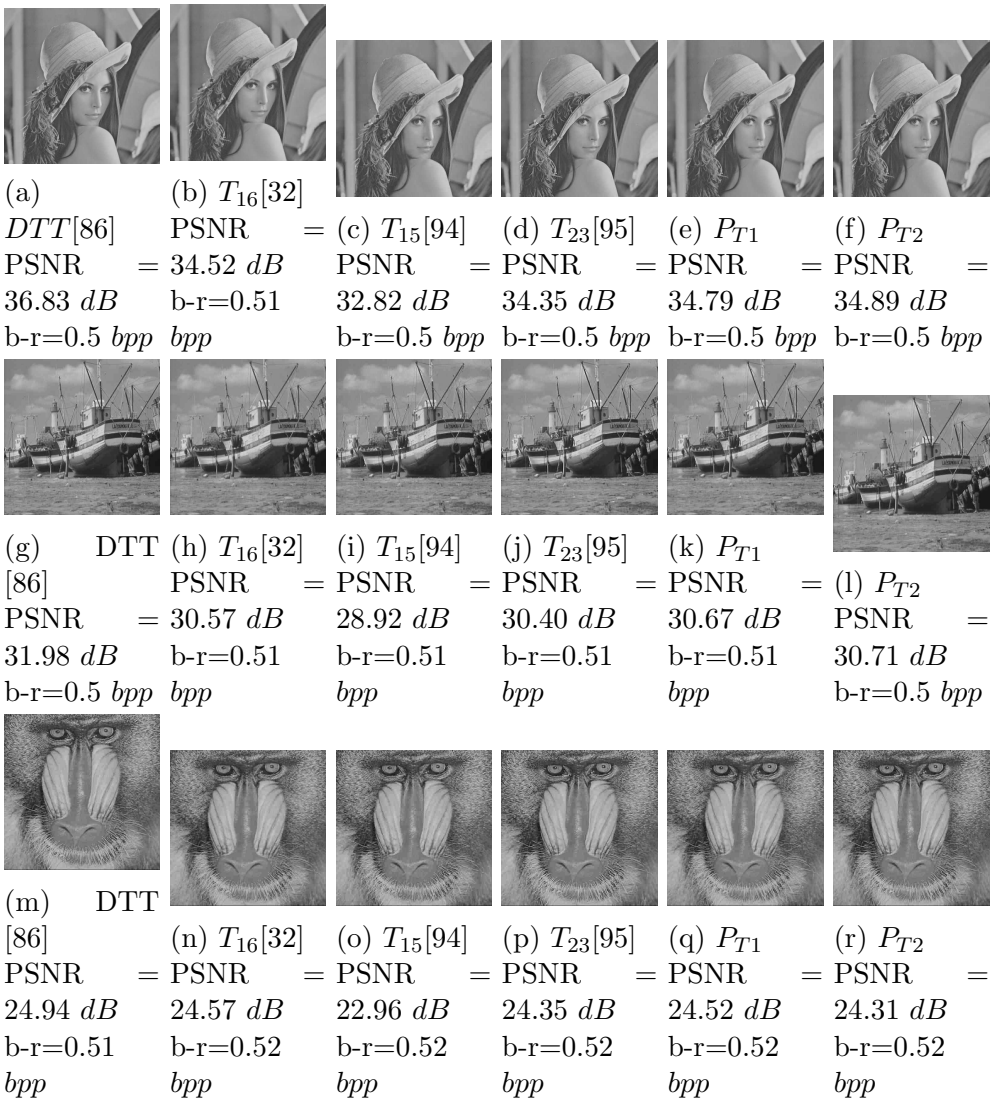

(a) SSIM vs bitrate

(b) PSNR vs bitrate

Figure 4.3: Average quality measures using the considered approximations vs bitrate.

For the visual evaluation, we selected three images from the dataset with different frequency content: 'Lena' with low frequencies, 'Baboon' with high frequencies, and 'Boat' with medium frequencies. We compressed these images at bit-rate around 0.5 bpp to assess the visual quality of the proposed compression methods. The results are presented in Fig. 4.4, which indicate that both proposed transforms, $P_{T2}$ and $P_{T3}$, outperform the other approx-

imations. For 'Lena' image, $P_{T2}$ showed an improvement of more than 0.37 dB compared to the best of the previous approximations (as seen in Fig. 4.4f vs. Fig. 4.4b). When comparing the best of the previous approximations, $O_{16}$ [32], to $P_{T2}$, the proposed approximation showed a gain of 0.27 dB (as seen in Fig. 4.4e vs Fig. 4.4b). For images with high-frequency elements, such as 'Baboon', all transformations showed similar quality, except $O_{15}$ [94], which had the lowest quality. $O_{16}$ [32] had a slight negligible gain of 0.05 dB compared to $P_{T2}$ (as seen in Fig. 4.4q vs Fig. 4.4n)



(a) *DTT*[86] PSNR = 36.83 *dB* b-r=0.5 *bpp*
(b) $T_{16}$[32] PSNR = 34.52 *dB* b-r=0.51 *bpp*
(c) $T_{15}$[94] PSNR = 32.82 *dB* b-r=0.5 *bpp*
(d) $T_{23}$[95] PSNR = 34.35 *dB* b-r=0.5 *bpp*
(e) $P_{T1}$ PSNR = 34.79 *dB* b-r=0.5 *bpp*
(f) $P_{T2}$ PSNR = 34.89 *dB* b-r=0.5 *bpp*
(g) DTT [86] PSNR = 31.98 *dB* b-r=0.5 *bpp*
(h) $T_{16}$[32] PSNR = 30.57 *dB* b-r=0.51 *bpp*
(i) $T_{15}$[94] PSNR = 28.92 *dB* b-r=0.51 *bpp*
(j) $T_{23}$[95] PSNR = 30.40 *dB* b-r=0.51 *bpp*
(k) $P_{T1}$ PSNR = 30.67 *dB* b-r=0.51 *bpp*
(l) $P_{T2}$ PSNR = 30.71 *dB* b-r=0.5 *bpp*
(m) DTT [86] PSNR = 24.94 *dB* b-r=0.51 *bpp*
(n) $T_{16}$[32] PSNR = 24.57 *dB* b-r=0.52 *bpp*
(o) $T_{15}$[94] PSNR = 22.96 *dB* b-r=0.52 *bpp*
(p) $T_{23}$[95] PSNR = 24.35 *dB* b-r=0.52 *bpp*
(q) $P_{T1}$ PSNR = 24.52 *dB* b-r=0.52 *bpp*
(r) $P_{T2}$ PSNR = 24.31 *dB* b-r=0.52 *bpp*

Figure 4.4: Reconstructed images for bit-rate $\approx 0.5$ *bpp*.

The results for a bit-rate of 1.92 bpp are presented in Fig. 4.5. The pro-

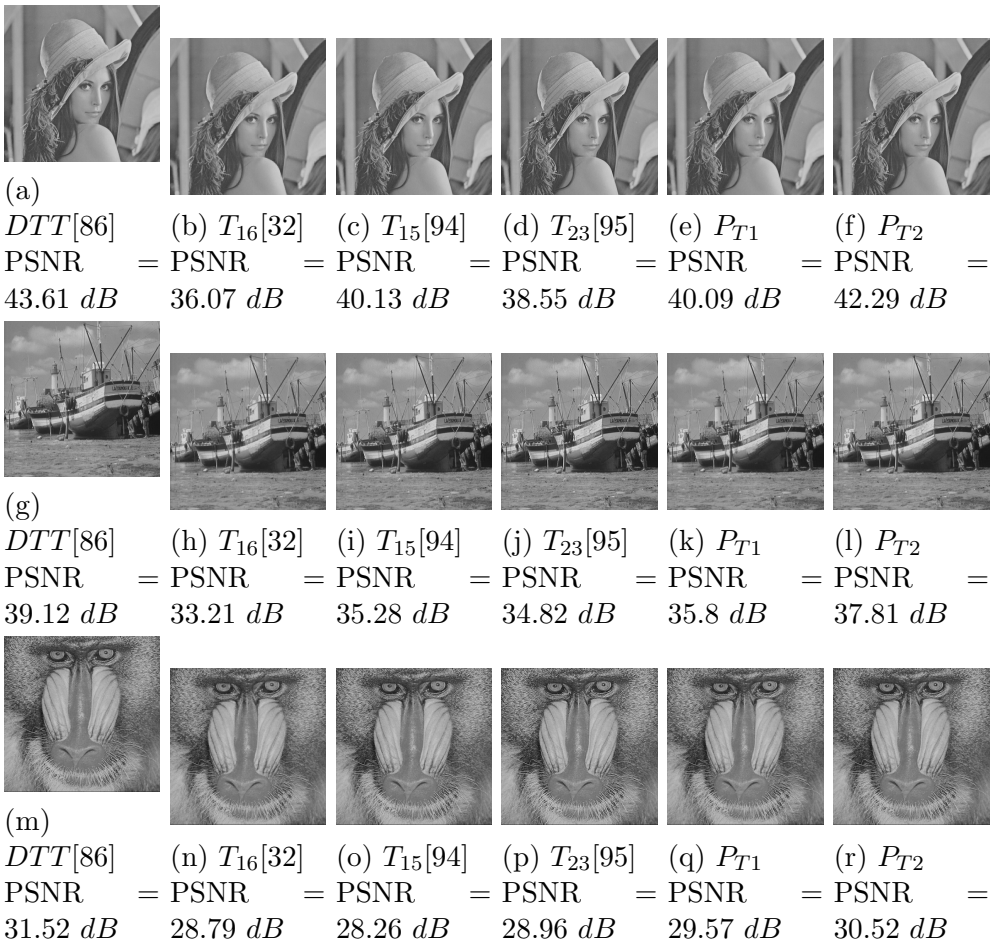



Figure 4.5: Reconstructed images for bit-rate $\approx 1.92$ *bpp*.

posed transformations exhibit superior quality compared to other approximations, with $P_{T3}$ providing an improvement of up to 2 dB in most images. In comparison to $O_{15}$, $P_{T2}$ exhibits a marginal decrease of 0.04 dB for 'Lena,' along with improvements of 0.52 dB and 0.78 dB for 'Boat' and 'Baboon', respectively. Compared to the exact DTT [86], the best-proposed transformation, $P_{T3}$, demonstrates a loss of 1.32 dB, 1.31 dB, and 1 dB for 'Lena', 'Boat', and 'Baboon', respectively. Nonetheless, the proposed transformations exhibit remarkably low computational complexity, requiring 69.86% fewer operations, as shown in Table 4.7.

## 4.7 Conclusion

This chapter presents novel approximations of the exact DTT, designed to exhibit minimal deviation from orthogonality and demonstrate efficacy in image processing applications such as image compression. These new approximations approach near-orthogonality and each is characterized by a computationally efficient algorithm requiring only 19 additions and 3 bit-shift operations. Notably, the forward and inverse transformations utilize the same algorithm, resulting in reduced arithmetic complexity compared to previous DTT approximations [94, 32] and the exact DTT [86], with reductions of 22.0%, 26.67%, and 69.86%, respectively.

Moreover, the proposed approximations not only yield superior image quality but also demonstrate enhanced hardware efficiency, as evidenced by their lower power consumption in FPGA implementations. These findings underscore the significant potential of the proposed DTT approximations for applications in mobile image processing and battery-powered devices.

Future research could focus on further improving the approximation's efficiency by enhancing the orthogonality of the transformation matrix. This could involve reducing additional deviations or reallocating them to high-frequency elements. Additionally, while current DCT/DTT approximations utilize the standard JPEG quantization matrix, future work could explore designing customized quantization matrices tailored for each approximation to further optimize performance.

# Chapter 5

# Optimizing Image Quality through Low-Complexity Implementation of DTT for Efficient image Compression

## Introduction

Discrete transforms [59, 105] play a pivotal role in various signal processing applications, particularly in data compression, where they map input data to a smaller set of output data. Consequently, the DCT finds wide application in various transform-based methods, including standards and DSP techniques such as JPEG still image compression [43, 106] and video compression codecs [107, 44]. Moreover, transforms represent one of the most computationally demanding components in modern video encoders [108], primarily due to the substantial computational resources required for arithmetic operations. Consequently, several attempts have been made to reduce the number of arithmetic operations by approximating the exact DCT, with successful efforts achieving as low as 14 additions [54, 55, 58].

Recently, the DTT [86] has emerged as a lower complexity discrete transform for picture coding, exhibiting characteristics similar to the DCT. However, the DTT has a lower number of approximations compared to the DCT, with only three approximations reported in the literature [94, 32, 95]. Among these approximations, the lowest number of operations is 23, with 19 additions presented in [95].

While DTT is an orthogonal transform, its approximations are not orthogonal, leading to two options. First, one can use the transpose matrix in

the backward transform, as done in [32, 95]. This option allows for the use of a single architecture for both forward and backward transforms but results in a loss of image quality. Alternatively, one can use the inverse matrix, as demonstrated in [94]. This option provides the best possible quality offered by the transform kernel but requires more hardware resources as two different architectures are needed.

However, the slight deviation from orthogonality offers certain advantages. Firstly, it leads to lower errors when using the transpose matrix since the transpose is almost equal to the inverse. Secondly, the inverse kernel has a complexity that is almost the same as the kernel itself. The approximation with the lowest deviation from orthogonality is presented in [95], where the transpose of the kernel is used as an approximation to its inverse. In this chapter, we present the inverse of the kernel and implement both variants within a single architecture. The proposed method resolves the issue of hardware complexity and demonstrates superior image quality compared to previous approximations.

## 5.1 proposed 8 points transform

The transformation proposed in [95] is based on the matrix and its transpose. However, in this study, we propose a new transformation that utilizes the inverse of the matrix instead of the transpose. The matrix kernel of the transformation $T_{p1}$ [95] exhibits non-orthogonality, indicating that its inverse does not equal its transpose. Consequently, implementing the transformation with two separate architectures is required, leading to increased hardware resource consumption.

It is worth noting that while the kernel of $T_{p1}$ deviates slightly from orthogonality, the deviations are considerably lower compared to other transformations such as those presented in [94, 32]. Notably, these deviations primarily occur in rows 3, 5, and 7 of the matrix. Therefore, we can merge the two architectures (for the kernel and its inverse) into a single architecture, retaining most of the rows while only modifying rows 3, 5, and 7. The inverse matrix of $T_{p1}$ can be defined as follows:

$$T_{p1}^{-1} = T_{p1i} \cdot D_4, \tag{5.1}$$

with:

$$T_{p1i0} = \begin{pmatrix} 1 & -2 & 3 & -1 & 3 & 0 & 1 & 0 \\ 1 & -1 & -1 & 1 & -9 & 1 & -3 & 0 \\ 1 & -1 & -1 & 1 & -1 & -1 & 5 & 0 \\ 1 & 0 & -1 & 0 & 7 & 0 & -3 & 1 \\ 1 & 0 & -1 & 0 & 7 & 0 & -3 & -1 \\ 1 & 1 & -1 & -1 & -1 & 1 & 5 & 0 \\ 1 & 1 & -1 & -1 & -9 & -1 & -3 & 0 \\ 1 & 2 & 3 & 1 & 3 & 0 & 1 & 0 \end{pmatrix}, \tag{5.2}$$

$$D_{1i} = diag(\frac{1}{8}, \frac{1}{12}, \frac{1}{16}, \frac{1}{6}, \frac{1}{32}, \frac{1}{4}, \frac{1}{32}, \frac{1}{2}). \tag{5.3}$$

On the contrary, the output of rows 3, 5, and 7 in the inverse kernel exhibits larger bit sizes compared to the original kernel. Consequently, it becomes necessary to reconcile these differences. One straightforward approach is to use the largest size as a common bit size, but this would incur a higher hardware resource requirement. Another solution is to reduce the bit size of the output in the inverse kernel (11 bits as maximum). This leads to a modified kernel, denoted as $T_{p1i}$, which is presented in equation (5.4).

$$T_{p1i} = \begin{pmatrix} 1 & -2 & \frac{3}{2} & -1 & \frac{3}{8} & 0 & \frac{1}{4} & 0 \\ 1 & -1 & -\frac{1}{2} & 1 & -\frac{9}{8} & 1 & -\frac{3}{4} & 0 \\ 1 & -1 & -\frac{1}{2} & 1 & -\frac{1}{8} & -1 & \frac{5}{4} & 0 \\ 1 & 0 & -\frac{1}{2} & 0 & \frac{7}{8} & 0 & -\frac{3}{4} & 1 \\ 1 & 0 & -\frac{1}{2} & 0 & \frac{7}{8} & 0 & -\frac{3}{4} & -1 \\ 1 & 1 & -\frac{1}{2} & -1 & -\frac{1}{8} & 1 & \frac{5}{4} & 0 \\ 1 & 1 & -\frac{1}{2} & -1 & -\frac{9}{8} & -1 & -\frac{3}{4} & 0 \\ 1 & 2 & \frac{3}{2} & 1 & \frac{3}{8} & 0 & \frac{1}{4} & 0 \end{pmatrix} \tag{5.4}$$

An alternative proposition involves further reducing the size of the output to a maximum of 10 bits instead of 11 bits. This proposition offers two advantages. Firstly, it is expected to result in lower hardware resource requirements compared to the approach using 11 bits. Secondly, by applying the bit reduction prior to addition, power consumption can be effectively reduced. Therefore, the proposed transformations can be summarized as follows:

$$P_{T4} \longrightarrow Y_1 = T_{p1} \cdot X \cdot T_{p1i} \cdot D_4, \tag{5.5}$$

$$P_{T5} \longrightarrow Y_2 = T_{p4} \cdot X \cdot T_{p4i} \cdot D_5, \tag{5.6}$$

with:
$X$: 8 points input vector.
$Y$: The transform domain of the input $X$.

$T_{p1}$: Defined in equation (3.6) [95].
$T_{p1i}$: Defined in equation (5.4).

$$D_4 = diag(\frac{1}{8}, \frac{1}{12}, \frac{1}{8}, \frac{1}{6}, \frac{1}{4}, \frac{1}{4}, \frac{1}{8}, \frac{1}{2}), \tag{5.7}$$

$$T_{p4} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ -1 & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 1 \\ 1 & 0 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 1 \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ 0 & -1 & 0 & 1 & 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \end{pmatrix}, \tag{5.8}$$

$$T_{p4i} = \begin{pmatrix} \frac{1}{2} & -1 & \frac{3}{4} & -\frac{1}{2} & \frac{3}{16} & 0 & \frac{1}{8} & 0 \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{4} & \frac{1}{2} & -\frac{9}{16} & 1 & -\frac{3}{8} & 0 \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{4} & \frac{1}{2} & -\frac{1}{16} & -1 & \frac{5}{8} & 0 \\ \frac{1}{2} & 0 & -\frac{1}{4} & 0 & \frac{7}{16} & 0 & -\frac{3}{8} & 1 \\ \frac{1}{2} & 0 & -\frac{1}{4} & 0 & \frac{7}{16} & 0 & -\frac{3}{8} & -1 \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{4} & -\frac{1}{2} & -\frac{1}{16} & 1 & \frac{5}{8} & 0 \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{4} & -\frac{1}{2} & -\frac{9}{16} & -1 & -\frac{3}{8} & 0 \\ \frac{1}{2} & 1 & \frac{3}{4} & \frac{1}{2} & \frac{3}{16} & 0 & \frac{1}{8} & 0 \end{pmatrix}, \tag{5.9}$$

$$D_5 = diag(\frac{1}{2}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{2}, \frac{1}{4}, \frac{1}{2}, \frac{1}{2}), \tag{5.10}$$

It is crucial to highlight that $Y_1$ is not equal to $Y_2$, but there exists a similarity between $P_{T4}$ and $P_{T5}$. Specifically, we observe that:

$$\begin{aligned} P_{T5} \longrightarrow Y_2 &= T_{p4} \cdot X \cdot T_{p4i} \cdot D_5 \\ &= \alpha_1 \cdot (T_{p1} \cdot X \cdot T_{p1i}) \cdot \alpha_2 \cdot D_5, \end{aligned} \tag{5.11}$$

where: $\alpha_1 = diag(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1, 1, \frac{1}{2}, 1)$
and $\alpha_2 = diag(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1, \frac{1}{2}, 1)$.

The recovery of the reconstructed block $X$ is achieved through the inverse transformation, as outlined below:

$$P_{T4}^{-1} \longrightarrow X_1 = T_{p1i} \cdot D_4 \cdot Y_1 \cdot T_{p1}, \tag{5.12}$$

$$P_{T5}^{-1} \longrightarrow X_2 = T_{p4i} \cdot D_5 \cdot Y_2 \cdot T_{p4}, \tag{5.13}$$

here, $X_1$ and $X_2$ represent the reconstructed blocks, serving as approximations to the input block $X$.

## 5.2 Performance assessment

In this section, an evaluation is conducted based on the coding gain ($C_g$) [109], transform efficiency ($\eta$) [59, 109], and deviation from orthogonality ($\delta$) [109] of the $8 \times 8$ transformation kernels. The results presented in Table 5.1 demonstrate that $O_{16}$ exhibits the most favorable outcomes in terms of $C_g$, $\eta$, and mean squared error (MSE) [59, 109]. Notably, the approximation $T_{p1}$ ranks second in terms of $C_g$, while demonstrating the best results in terms of $\delta$ compared to previous approximations. These findings establish that $O_{16}$ showcases superior coding performance, albeit accompanied by errors arising from its high $\delta$. Conversely, $T_{p1}$ offers acceptable coding performance with a reduced level of error.

Nevertheless, while the proposed transformations exhibit a lower $C_g$ when compared to $T_{p1}$, they offer superior compression efficiency owing to their inherent reversibility facilitated by the utilization of the kernel and its inverse. Furthermore, these proposed transformations demonstrate elevated values of $\eta$, underscoring their enhanced compression efficiency. However, among the array of proposed transformations, it is noteworthy that $P_{T5}$ stands out, showcasing higher values of both $C_g$ and $\eta$ in comparison to $P_{T4}$.

Table 5.1: Performance assessment.

| Method | $C_g[db]$ | $\eta$ | $MSE$ | $\delta$ |
|---|---|---|---|---|
| DTT [86] | 8.68 | 92.86 | - | 0 |
| $O_{16}$ [32]. | 8.57 | 89.52 | 0.0022 | 0.024 |
| $T_{p1}$ [95] | 7.85 | 85.77 | 0.0115 | 0.014 |
| $P_{T4}$ | 7.34 | 85.87 | 0.0115 | 0.014 |
| $P_{T5}$ | 7.64 | 86.35 | 0.0115 | 0.014 |

It is important to highlight that the similarity in MSE and $\delta$ values between the proposed approximations and $T_{p1}$ can be attributed to the fact that the formulas for MSE and $\delta$ exclusively employ the matrix utilized in the forward transform and its transpose. This observation underscores that all the aforementioned approximations pertain to the identical matrix when subjected to the transpose operation.

## 5.3 Arithmetic complexity

This section provides an analysis of the arithmetic complexity of the proposed transformations in comparison to other existing transformations [32, 95]. Table 5.2 presents the arithmetic complexity of the proposed transformations

as well as those defined in [32, 95]. The results indicate that the proposed transformations rank as the second-best in terms of arithmetic complexity, following the performance of $T_{p1}$ [95].

In particular, it is worth noting that $P_{T4}$ and $P_{T5}$ exhibit an equal number of additions, while $P_{T5}$ entails a higher number of bit-shift operations compared to $P_{T4}$. It is important to emphasize that bit-shift operations primarily involve wiring and thus have a minimal impact on energy consumption. Furthermore, $P_{T5}$ eliminates the need for 11-bit additions and replaces them with 10-bit additions, resulting in a reduction of 11-bit additions to 0.

Overall, these findings highlight the favorable arithmetic complexity of the proposed transformations, with $P_{T5}$ introducing significant improvements by reducing the number of 11-bit additions while maintaining the overall computational efficiency.

Table 5.2: Arithmetic complexity comparison for 8-point DTT approximations.

| | 8-point 1D | | |
|---|---|---|---|
| Approximation | adds | b-shifts | Total |
| DTT[86] Forward | 44 | 29 | 73 |
| DTT[86] Inverse | 44 | 29 | 73 |
| **Total** | **88** | **58** | **146** |
| $O_{16}$ [32]. Forward | 24 | 6 | 30 |
| $O_{16}$ [32]. Inverse | 24 | 6 | 30 |
| **Total** | **48** | **12** | **60** |
| $T_{p1}$ [95] Forward | 19 | 3 | 22 |
| $T_{p1}$ [95] Inverse | 19 | 3 | 22 |
| **Total** | **38** | **6** | **44** |
| $P_{T4}$ Forward | 19 | 3 | 22 |
| $P_{T4}$ Inverse | 22 | 6 | 28 |
| **Total** | **41** | **9** | **50** |
| $P_{T5}$ Forward | 19 | 6 | 25 |
| $P_{T5}$ Inverse | 22 | 10 | 32 |
| **Total** | **41** | **16** | **57** |

Figures 5.1 and 5.2 represent the SFG of the data, which provide visual confirmation of the number of operations involved in the proposed $P_{T4}$ and $P_{T5}$ transformations, respectively. In these figures, the black wires represent connections that are common for both forward and backward transforms, the red wires represent connections specific to the forward transform, and the blue wires represent connections specific to the backward transform. The

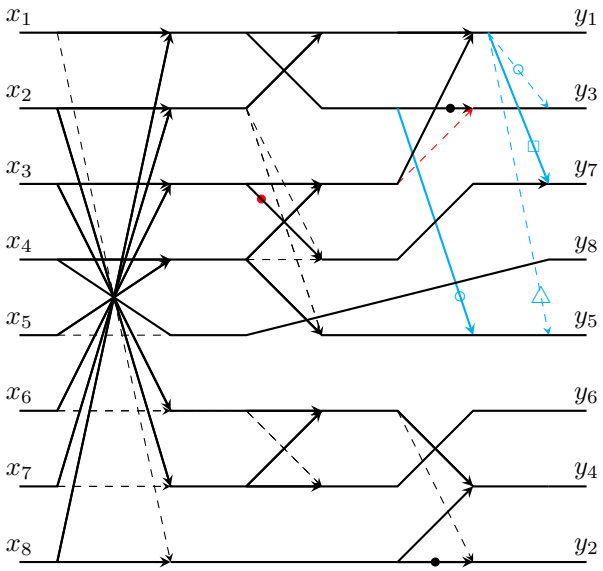



Figure 5.1: Signal graph flow of data for the proposed $P_{T4}$ transformation. Where •, ○, □ and △ present 1-bit right-bit-shift, 1-bit left-bit-shift, 2-bits left-bit-shift and 3-bits left-bit-shift, respectively.

SFGs offer a comprehensive overview of the data flow and enable a better understanding of the computational processes involved in the transformations.

## 5.4 Hardware implementation

This section discusses the hardware implementation of the proposed algorithms in a FPGA platform. FPGA technology offers a flexible and reconfigurable hardware solution that is well-suited for real-time signal processing applications. In this section, we present the details of the FPGA implementation, including the design considerations, resource utilization, and performance evaluation. The goal is to demonstrate the feasibility and effectiveness of the proposed algorithms in a hardware implementation, showcasing their potential for efficient and low-power processing in practical applications.

The results of the implementation are tabulated in Table 5.3, showcasing that the proposed approximations, denoted as $P_{T4}$ and $P_{T5}$, demand reasonable hardware resources when compared to $O_{16}$ and $T_{p1}$. Notably, $P_{T5}$ necessitates 16 additional slice registers and 37 more slice LUTs, while manifesting a reduction of 31 in FF when juxtaposed with $T_{p1}$. In comparison with $O_{16}$, $P_{T5}$ exhibits a decline in resource utilization, specifically a decrement of

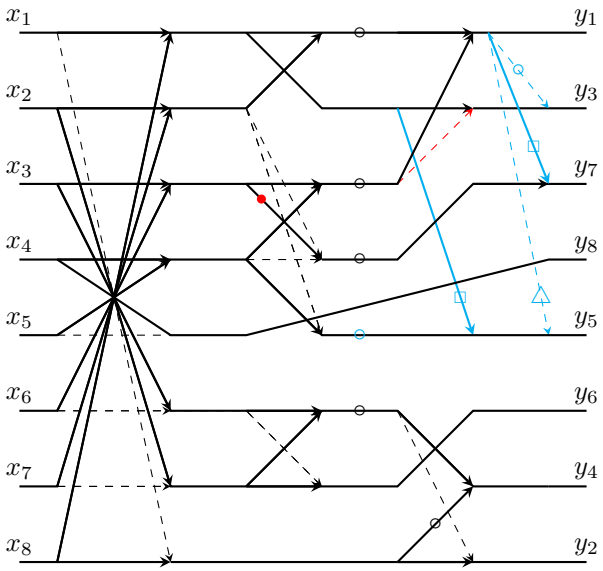



Figure 5.2: Signal graph flow of data for the proposed $P_{T5}$ transformation. Where •, ∘, □ and △ present 1-bit right-bit-shift, 1-bit left-bit-shift, 2-bits left-bit-shift and 3-bits left-bit-shift, respectively.

Table 5.3: Hardware resource consumption using Xilinx Virtex-6 XC6VSX475T-1FF1759

| | $O_{16}$ [32] | $T_{p1}$ [95] | $P_{T4}$ | $P_{T5}$ |
|---|---|---|---|---|
| Slice Registers | 212 | 188 | 216 | 204 |
| Slice LUTs | 248 | 177 | 223 | 214 |
| FF | 316 | 262 | 243 | 231 |
| Tcpd (ns) | 1.924 | 1.908 | 2.331 | 2.314 |
| Fmax (MHz) | 519.7 | 524.1 | 428.9 | 432.1 |
| Power (W) | 4.686 | 4.63 | 4.579 | 4.564 |

8 slice registers, 34 slice LUTs, and 85 FF.

Furthermore, the proposed approximations exhibit diminished power consumption compared to their precursors, revealing a noteworthy reduction of 66 mW and 122 mW for the superior approximation, $P_{T5}$, in contrast to $T_{p1}$ and $O_{16}$, respectively. It is worth noting, however, that the proposed approximations tend to demonstrate lower frequency (Fmax) and prolonged delay (Tcpd) as compared to the previous approximations. Nonetheless, this trade-off is counterbalanced by the pronounced advantage of enhanced compression efficiency, which will be expounded upon in the subsequent section.

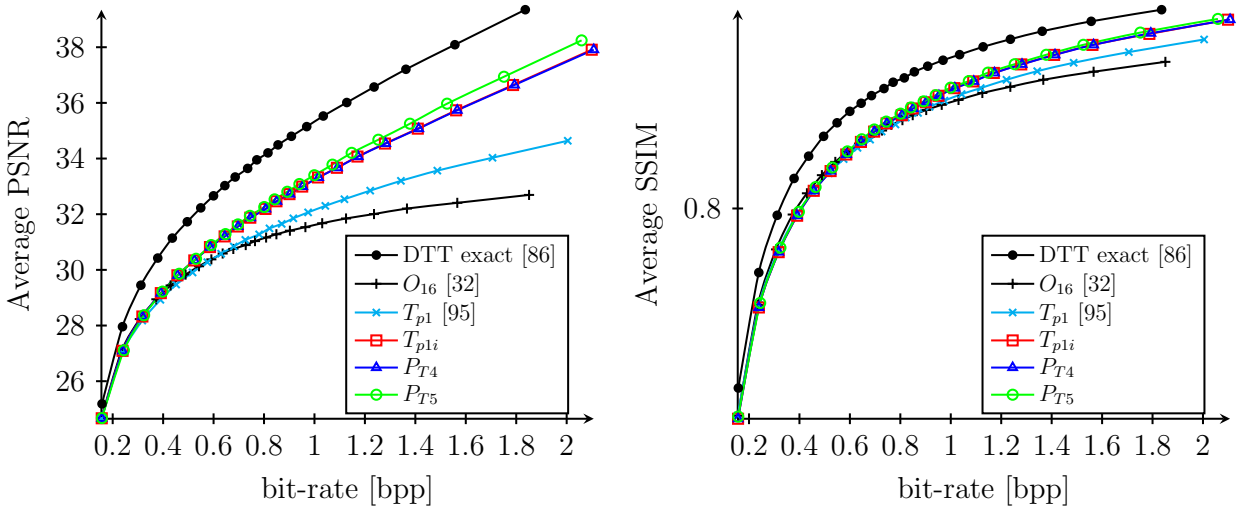



Figure 5.3: Quality evaluation of the proposed transformations compared to [86, 32, 95].

## 5.5 Image applications

In this study, the proposed approximations were integrated into a JPEG compression chain, along with the transformations of [86, 32, 95]. To evaluate the performance of the algorithm, 47 grayscale images from a public dataset [66] were used. The decompressed images obtained using the inverse 2-D transforms were compared to the original images. The quality of the decompressed images was assessed using two well-established metrics: the PSNR [61] and the SSIM [67]. These metrics provide objective measures of image quality, with higher PSNR and SSIM values indicating better image fidelity and similarity to the original images, respectively. By employing these metrics, the study evaluated the effectiveness of the proposed algorithm in preserving image quality during the JPEG compression process.

### 5.5.1 Results and discussion

The image quality assessment results, presented in Fig. 5.3, demonstrate the superiority of the proposed approximations compared to other existing approximations [32, 95] in terms of PSNR. For bit-rates lower than 0.4 bits per pixel (bpp), the reconstructed images using the proposed approximations exhibit similar quality to the other approximations. However, at higher bit-rates, the proposed approximations outperform the others, yielding a gain of over 2 dB in PSNR.

Regarding the SSIM, the results of the proposed approximation align closely with the other approximations [32, 95]. There is a negligible loss in SSIM for bit-rates below 0.5 bpp, and a slight gain for bit-rates exceeding 0.6 bpp.

These findings indicate that the proposed approximations achieve superior image quality, particularly at higher bit-rates, while maintaining comparable performance to other approximations in terms of PSNR and SSIM. This demonstrates the efficacy of the proposed approach in enhancing the quality of reconstructed images in the context of JPEG compression.

## 5.6 Visual evaluation

For the purpose of visual evaluation, an image containing medium-frequency elements, denoted as 'Boat,' has been meticulously chosen from a publicly available database [66]. The process of image reconstruction entailed the application of the inverse transformation utilizing the approximations elucidated in Table 5.2. The outcomes of this endeavor are meticulously exhibited in Fig. 5.4. All of the images underwent compression at a lower bit-rate of 0.5 bpp, a value judiciously selected to represent a pragmatic choice for embedded systems, signifying a substantial 90% reduction in image size.

| (a) $DTT$[86] | (b) $O_{16}$[32] | (c) $T_{p1}$[95] | (d) $P_{T4}$ | (e) $P_{T5}$ |
|---|---|---|---|---|
| PSNR = 31.98 *dB* | PSNR = 30.52 *dB* | PSNR = 30.32 *dB* | PSNR = 30.71 *dB* | PSNR = 30.51 *dB* |
| b-r=0.5 *bpp* | b-r=0.5 *bpp* | b-r=0.5 *bpp* | b-r=0.5 *bpp* | b-r=0.5 *bpp* |

Figure 5.4: Reconstructed images at bit-rate equals to 0.5 bpp.

The insights garnered from the results depicted in Fig. 5.3 illuminate the commendable performance of the proposed approximations at higher bit-rates. Notably, the approximation $T_{p1}$ [95] does not yield optimal image quality results in comparison to $O_{16}$ at lower bit-rates. However, its enhanced iterations, $P_{T4}$ and $P_{T5}$, manifest more favorable outcomes in the same bit-rate domain. Further inspection of the reconstructed image at 0.5 bpp, as portrayed in Fig. 5.4, accentuates a significant enhancement in quality attributed to the proposed approximations. Particularly noteworthy is the case

of $P_{T4}$, which exhibits an impressive 0.39 dB and 0.19 dB improvement in image quality relative to $T_{p1}$ [95] and $O_{16}$ [32], respectively. In contrast, $P_{T5}$ showcases a 0.19 dB enhancement when compared to $T_{p1}$ [95], accompanied by a marginal 0.01 dB degradation when compared with $O_{16}$ [32].

Certainly, the proposed approximations, namely $P_{T4}$ and $P_{T5}$, not only yield notable enhancements in image quality but also manifest as reductions in power consumption. Specifically, $P_{T4}$ exhibits a decrease of 66 mW and 122 mW in power consumption as compared to $T_{p1}$ [95] and $O_{16}$ [32], respectively. Similarly, $P_{T5}$ showcases diminished power consumption, with reductions of 51 mW and 107 mW when compared to $T_{p1}$ [95] and $O_{16}$ [32], respectively, as meticulously detailed in Table 5.3.

## 5.7 Conclusion

In conclusion, this chapter introduces a novel approach to approximate the transform matrix by utilizing its inverse in a merged architecture. The proposed method demonstrates superior performance in terms of reconstructed image quality compared to existing approximations. It achieves an optimal balance between image quality, hardware resources, and energy consumption. By leveraging the advantages of the inverse kernel, this approach effectively eliminates errors caused by the transformation while significantly reducing hardware requirements and power consumption through the use of a single architecture. The results highlight a significant improvement in image quality while maintaining acceptable power consumption levels. Furthermore, this method can be applied to various quasi-orthogonal matrices found in the existing literature, extending its applicability to a wide range of scenarios.

# Chapter 6

# New Quantization Tables and Coefficient Ordering for Improving Efficiency of Approximate DCT and DTT Transformations

## Introduction

Image compression is a critical step in various applications, including telemedicine [110, 111, 22], remote sensing [112, 24, 113], Artificial intelligence , video compression codecs [44, 114, 115], and video conferencing. Transform-based image compression techniques have been widely used to achieve high compression rates while maintaining good image quality. These techniques convert the image from the spatial domain to a transform domain using various transformations, such as DTT [86], DCT [105], Discrete Sine Transform (DST), Walsh-Hadamard Transform (WHT) [116], and KLT[117].

In the realm of embedded systems and real-time applications, transform-based image compression adopts approximation techniques to mitigate the computational complexity of the encoder. Prior research has extensively explored approximating the DCT [29, 58, 54, 55, 47, 49] and the DTT [94, 32, 95]. Nevertheless, achieving a low-complexity approximation that strikes the optimal balance between image quality and bit-rate reduction remains an enduring challenge. It is important to note that the transformation step itself, particularly in the context of orthogonal kernels, is lossless. However, its impact on image quality hinges on coding performance and energy compaction

[95].

While the transformation kernel is a pivotal factor affecting image quality and bit-rate, it is by no means the sole determinant. Quantization, zig-zag patterns, and Huffman coding also wield considerable influence in this regard. Notably, studies have diligently sought to optimize Huffman tables [118, 119] and quantization tables for DCT-based JPEG [120, 121, 122, 123, 124], as well as for DTT-based JPEG [125, 126]. These investigations have underscored the pivotal role of quantization tables in compression performance. However, existing research primarily concentrates on optimizing quantization for DCT and DTT without specifically addressing their respective approximations.

Transform-based image compression leans on quantization to streamline the storage of coefficients. Yet, the standard quantization table, meticulously tailored for the DCT [127], may not consistently yield optimal compression performance for diverse approximations. This underscores the necessity of crafting new quantization tables tailored for each approximation, grounded in the unique characteristics of their coefficients. Furthermore, the sequence in which coefficients are ordered post-quantization can exert a profound influence on compression efficiency. The ubiquitous zig-zag order, frequently employed in image compression techniques, lacks the capability to discern the relative importance of each coefficient across various approximations.

This chapter reevaluate the conventional quantization table and zig-zag order applied to DCT and DTT approximations. The overarching objective is to enhance the capabilities of transform-based compression. The proposed methodology takes into account not only the approximation technique but also the quantization and zig-zag pattern to attain superior image quality while limiting the size of the compressed data.

Our approach entails the introduction of a modified quantization table that accords heightened significance to coefficients housing critical image data. This strategic adjustment will enhance image quality. Moreover, we introduce a coefficient reordering mechanism that prioritizes coefficients based on their relevance, thereby augmenting compression efficiency. A salient feature of our methods is their capacity to enhance performance without introducing supplementary arithmetic complexity, rendering them highly efficient. The contributions made by our work are as follows:

- An algorithm is introduced to generate optimal coefficient orders tailored to each specific approximation.
- A Novel coefficient ordering schemes are introduced, offering an alternative to the conventional zig-zag pattern, resulting in reduced bit-rates.

- Quantization tables are redefined based on the coefficient orders derived from the algorithm.

To assess the effectiveness of our proposed method, comprehensive evaluations were conducted across various transform-based compression techniques, encompassing DTT and DCT approximations [29, 54, 55, 58, 47, 49, 94, 95]. Consistently, our experimental findings underscore the superiority of our approach in terms of image quality when compared to the conventional quantization and zig-zag order methods. Additionally, a reduction in compressed data size is observed, indicative of enhanced image compression efficiency.

The subsequent sections of this chapter are structured as follows: Section 6.1 presents related works. Section 6.2 comprises an analysis of previous approximations, focusing on image quality and coding performance, followed by the description of the proposed algorithm. In Section 6.3, we conduct a performance evaluation of each approximation within the proposed method. The results of the image compression application, both in JPEG-like and JPEG, are detailed in Section 6.4 and Section 6.5, respectively. Finaly, Section 6.6 gives the conclusion and presents the outcomes of this study.

## 6.1 Related Work

In this section, we delve into the body of existing literature that contextualizes our research and provides valuable insights into the landscape of image compression, specifically focusing on transformation approximations and quantization tables. Firstly, we examine previous works pertaining to the efficient approximations of linear transformations, and second, we delve into the design of quantization tables.

### 6.1.1 Transform Approximations

BAS ($BAS_8$) [47] proposed an 8x8 transform matrix by selectively incorporating 0s and 1/2s into the $8 \times 8$ SDCT matrix described in [64]. $BAS_{11}$ [49] presents a novel eight-point transformation approach. It introduces an arbitrary parameter into the transformation matrix, building on the method in reference [47], and employs specific row permutations. This parameter substitution ensures orthogonality while row permutations boost energy compaction.

### 6.1.2 Quantization Tables

Quantization is a fundamental aspect of image compression, serving to reduce the number of bits required to encode images while preserving crucial image details. The standard quantization table [43] is meticulously designed to minimize distortion under the condition of maximum compression, with a primary focus on typical images and the visual characteristics of the average viewer. In this process, it aggressively reduces the representation of high-frequency coefficients, leading to data compression.

Moreover, alternative quantization tables have been developed, taking into account the characteristics of the human visual system (HVS). Notably, the $Q_{hvs}$ table [121], is based on the assumption of isotropic properties within the HVS. This quantization approach models the HVS as a nonlinear point transformation, followed by the application of a modulation transfer function.

The literature on image compression includes a plethora of quantization tables, each with its unique approach to balancing compression and image quality. These tables have been proposed in various studies, such as [120, 121, 122, 123, 124, 125, 126]. It's worth noting that our proposed method is versatile and can be applied effectively with any defined quantization table when used in conjunction with a specific approximation technique. This adaptability allows our method to be employed in various image compression scenarios and provides a comprehensive solution for improving the performance of different quantization tables when applied alongside appropriate approximations.

## 6.2 Proposed Method

### 6.2.1 Analysis of Previous Approximations

In this study, our main focus was to evaluate the quality of commonly used approximations, such as DCT and DTT, and compare them with selected transformations for potential improvement. Specifically, we analyzed the quality of the exact DTT and DCT and compared them with their respective approximations. Fig. 6.1 showcases the PSNR graphs for DTT and DCT, along with their corresponding approximations. The graphs of the exact DTT [86] and DCT represent the ideal shape, where the initial coefficients exhibit a significant increase in quality, followed by a gradual improvement in quality for the subsequent coefficients. However, deviations from this ideal shape are observed in the approximations presented in Fig. 6.1. Certain coefficients within these approximations exhibit remarkably high quality, underscoring their significance. However, it is noteworthy that these coefficients may not

be optimally positioned. For instance, a substantial improvement in quality is observed when coefficients such as 6, 10, 16, 21, and others are included. This observation underscores their importance in enhancing image quality.

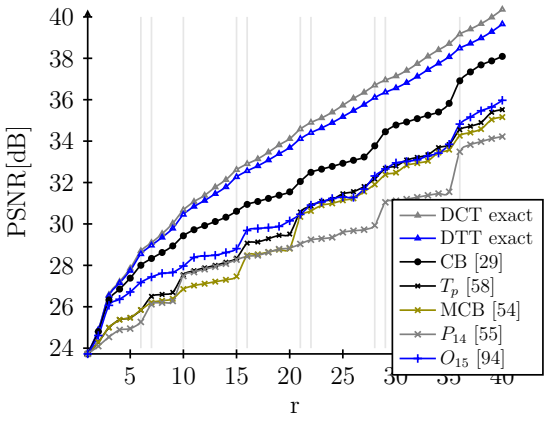



Figure 6.1: Average PSNR vs number of retained coefficients (r) of DCT (black) and DTT (blue) approximations.

### 6.2.2 Proposed Algorithm

To address the issue of achieving higher image quality in various approximations, including $T_{p1}$ [95], $O_{15}$ [94], $T_p$ [58], MCB [54], $\mathbf{P_{14}}$ [55], Cintra-Bayer approximation (CB) [29], and BAS series [47], we propose an algorithm that adjusts the quantization and zig-zag pattern for each approximation. The conventional JPEG quantization process and the zig-zag pattern may not be optimal for all approximations and may significantly affect important transform coefficients. Our algorithm aims to customize the quantization and zig-zag order for each specific approximation, thereby maximizing the overall quality of the transformed image. To achieve this, we have introduced Algorithm 2, which outlines the steps involved in determining the optimal coefficient order for obtaining the highest quality of the recovered image.

The algorithm takes into account 47 images obtained from a publicly available database [66]. To evaluate the impact of changing the $n^{th}$ coefficient on image quality. Each image is compressed using the selected approximation, and the algorithm examines the quality of the reconstructed images with only $n$ coefficients ($n = 2, 3, 4, ..., 64$). In this process, the previous $r$ coefficients are held fixed, with $r < n$, while the remaining coefficients ranging from $n+1$ to 64 are set to zero. By systematically varying the $n^{th}$ coefficient and assessing its impact on image quality, the algorithm identifies the coefficient that yields the greatest improvement. This coefficient is then added

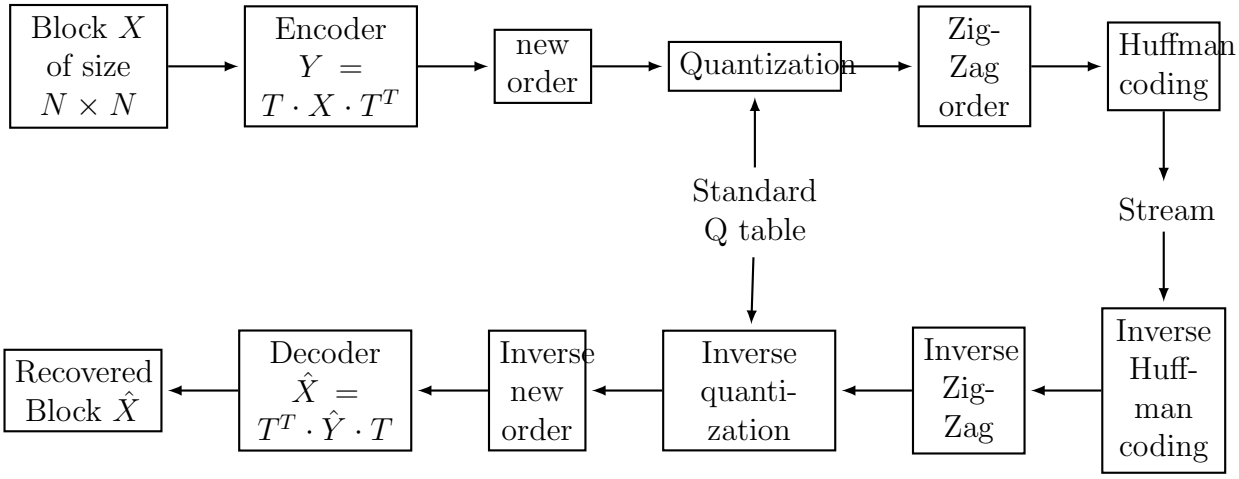



Figure 6.2: Proposed modification in the JPEG for study purposes

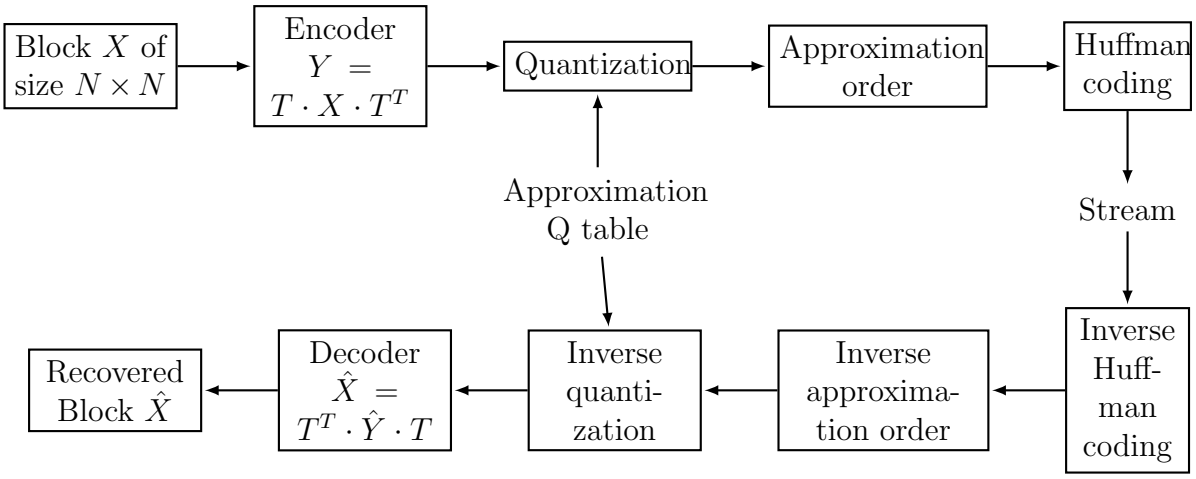



Figure 6.3: Proposed modification in the JPEG without adding complexity. Approximation $Q$ table and approximation order are presented in TABLE 6.1

to the fixed set of elements, thus refining the coefficient order for subsequent iterations.

**Algorithm 2** The proposed algorithm

**Require:** Dataset with a various set of images
**Ensure:** Optimized Order of Coefficients (OOC)

$images \leftarrow readImages()$ ▷ Read all the images in [66]
$I_n \leftarrow length(images)$
$OOC \leftarrow [1]$
$Cs \leftarrow [2, 3, ..., 64]$
**while** $length(Cs) > 0$ **do**
  **for** $i \leftarrow 1, 2, 3, ..., I_n$ **do**
    $encImg \leftarrow encode(images[i])$ ▷ Encode the image with the JPEG-LIKE compression.
    **for** $c \leftarrow 1, 2, 3, ..., length(Cs)$ **do**
      $C \leftarrow add(Coafs, Cs[c])$ ▷ Add the value of $Cs[c]$ into the array OOC
      $decImg \leftarrow decode(encImg, C)$ ▷ Decode the encoded image $encImg$ with only the coefficients in $C$
      $q[i, c] \leftarrow Eval(images[i], decImg)$ ▷ $Eval(a, b)$ return $PSNR(a, b) \times UQI(a, b)$.
    **end for**
  **end for**
  $Avg \leftarrow mean(q)$
  $maxIndex \leftarrow indexOfMax(Avg)$ ▷ Index of the maximum value in $Avg$
  $bestC \leftarrow Cs[maxIndex]$
  $OOC \leftarrow add(OOC, bestC)$ ▷ Add $bestC$ to OOC.
  $Cs \leftarrow remove(Cs, bestC)$ ▷ Remove $bestC$ from $Cs$.
**end while**
**return** OOC

### 6.2.3 Implementations in JPEG

Two methods are available for implementation. First, altering the coefficient order post-transform introduces complexity but is suitable for research, as shown in Fig. 6.2. One limitation is that it adds processing steps to the JPEG chain, increasing compression process complexity. For practicality, we opted for the initial implementation, which proved to be straightforward to implement. This approach involved introducing a "reorder" step in the

standard JPEG workflow, between coefficient calculation and quantization. This step rearranges the coefficients based on their importance while keeping the quantization table and zig-zag pattern unchanged. The second method involves adjusting the zig-zag order and quantization table in Fig. 6.3, which maintains standard JPEG complexity. Unlike the first method, it does not require additional steps. Nonetheless, it faces the limitation of consistent decoding across different systems. It is important to note that this study is designed for embedded systems and low-complexity devices that use custom approximations rather than the standard JPEG. Achieving compatibility necessitates adjustments in both the encoder and decoder to accommodate the proposed changes in the transform kernel, quantization table, and coefficient order. The new quantization table and coefficients order defined for each approximation are presented in Table 6.1.

### 6.2.4 Proposed Quantization and Coefficients Order

In the field of image compression, adjusting the quantization of the coefficients can significantly impact the quality of the recovered images. However, this process can also cause a slight increase in the output data size due to the change in the order of coefficients based on their importance. Surprisingly, the last coefficient that mostly becomes zero due to quantization may be more important than the first few coefficients. To address this issue, we propose a novel approach to changing the standard zig-zag pattern to reflect the importance of the coefficients in each approximation, allowing us to achieve a lower image bit-rate while maintaining high image quality.

All existing approximations in the literature commonly employ the standard quantization table and zig-zag order, as shown in Table 6.1. However, in this study, we have introduced a novel approach by modifying both the coefficients order (the result of Algorithm 2) and the corresponding quantization table for each examined approximation, as presented in Table 6.1. The proposed quantization table is derived from the standard table and the results of Algorithm 2, with coefficients adjusted according to their respective importance. Notably, in the proposed table, coefficients $r2$ and $r3$ are swapped compared to the standard table, indicating a higher importance of the first coefficient in the second row. Consequently, these coefficient swaps are reflected in the corresponding quantization table. This approach ensures that the quantization process aligns with the modified order of the coefficients, optimizing the compression performance.

Our method involves rearranging coefficients based on their significance before quantization, resulting in similar coefficients being obtained at different locations. By adhering to standard coefficients and preserving their

Table 6.1: Standard and new quantization table and coefficient order (zigzag) for each approximation.

| Transform | Coefficient order | Quantization table |
|---|---|---|
| Standard | $\begin{bmatrix} r1 & r2 & r6 & r7 & r15 & r16 & r28 & r29 \\ r3 & r5 & r8 & r14 & r17 & r27 & r30 & r43 \\ r4 & r9 & r13 & r18 & r26 & r31 & r42 & r44 \\ r10 & r12 & r19 & r25 & r32 & r41 & r45 & r54 \\ r11 & r20 & r24 & r33 & r40 & r46 & r53 & r55 \\ r21 & r23 & r34 & r39 & r47 & r52 & r56 & r61 \\ r22 & r35 & r38 & r48 & r51 & r57 & r60 & r62 \\ r36 & r37 & r49 & r50 & r58 & r59 & r63 & r64 \end{bmatrix}$ | $\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 65 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$ |
| CB [29] | $\begin{bmatrix} r1 & r3 & r5 & r8 & r13 & r14 & r16 & r17 \\ r2 & r6 & r10 & r20 & r26 & r29 & r27 & r33 \\ r4 & r9 & r21 & r25 & r34 & r40 & r36 & r44 \\ r7 & r18 & r24 & r31 & r37 & r46 & r43 & r52 \\ r11 & r22 & r32 & r38 & r45 & r53 & r49 & r59 \\ r12 & r28 & r39 & r47 & r54 & r58 & r56 & r63 \\ r15 & r23 & r35 & r41 & r48 & r55 & r51 & r61 \\ r19 & r30 & r42 & r50 & r57 & r62 & r60 & r64 \end{bmatrix}$ | $\begin{bmatrix} 16 & 12 & 12 & 14 & 16 & 19 & 40 & 26 \\ 11 & 10 & 14 & 22 & 40 & 61 & 58 & 56 \\ 14 & 13 & 24 & 29 & 55 & 68 & 72 & 56 \\ 16 & 24 & 37 & 57 & 92 & 109 & 55 & 104 \\ 18 & 49 & 51 & 78 & 80 & 103 & 95 & 100 \\ 17 & 51 & 65 & 81 & 62 & 112 & 113 & 103 \\ 24 & 35 & 64 & 87 & 87 & 77 & 103 & 92 \\ 22 & 60 & 69 & 98 & 121 & 101 & 120 & 99 \end{bmatrix}$ |
| $T_p$ [58] | $\begin{bmatrix} r1 & r3 & r7 & r5 & r11 & r9 & r16 & r14 \\ r2 & r12 & r20 & r17 & r37 & r26 & r44 & r46 \\ r6 & r19 & r21 & r23 & r34 & r28 & r42 & r48 \\ r4 & r18 & r24 & r22 & r43 & r29 & r47 & r50 \\ r10 & r33 & r32 & r35 & r56 & r45 & r58 & r60 \\ r8 & r25 & r27 & r30 & r49 & r31 & r53 & r54 \\ r13 & r36 & r38 & r39 & r57 & r52 & r61 & r62 \\ r15 & r41 & r40 & r51 & r59 & r55 & r63 & r64 \end{bmatrix}$ | $\begin{bmatrix} 16 & 12 & 16 & 12 & 18 & 13 & 40 & 19 \\ 11 & 17 & 22 & 26 & 92 & 40 & 56 & 109 \\ 10 & 22 & 24 & 35 & 55 & 51 & 69 & 87 \\ 14 & 24 & 37 & 49 & 55 & 61 & 81 & 98 \\ 14 & 56 & 51 & 64 & 113 & 80 & 112 & 120 \\ 14 & 29 & 58 & 60 & 95 & 57 & 103 & 62 \\ 16 & 72 & 78 & 65 & 121 & 104 & 92 & 101 \\ 24 & 87 & 68 & 103 & 100 & 77 & 103 & 99 \end{bmatrix}$ |
| $O_{15}$ [94] | $\begin{bmatrix} r1 & r3 & r5 & r9 & r13 & r7 & r14 & r19 \\ r2 & r6 & r12 & r18 & r26 & r17 & r27 & r38 \\ r4 & r21 & r42 & r45 & r62 & r31 & r64 & r56 \\ r10 & r20 & r34 & r36 & r49 & r29 & r50 & r54 \\ r8 & r22 & r43 & r46 & r61 & r32 & r63 & r57 \\ r15 & r24 & r30 & r35 & r39 & r28 & r40 & r53 \\ r11 & r23 & r44 & r47 & r59 & r33 & r60 & r58 \\ r16 & r25 & r41 & r48 & r51 & r37 & r52 & r55 \end{bmatrix}$ | $\begin{bmatrix} 16 & 12 & 12 & 13 & 16 & 16 & 19 & 22 \\ 11 & 10 & 17 & 24 & 40 & 26 & 58 & 78 \\ 14 & 24 & 69 & 80 & 101 & 57 & 99 & 113 \\ 14 & 22 & 55 & 72 & 95 & 61 & 98 & 62 \\ 14 & 49 & 55 & 109 & 92 & 51 & 103 & 121 \\ 24 & 37 & 60 & 64 & 65 & 51 & 68 & 103 \\ 18 & 35 & 56 & 81 & 100 & 56 & 120 & 112 \\ 40 & 29 & 87 & 87 & 103 & 92 & 104 & 77 \end{bmatrix}$ |
| $T_{p1}$ [95] | $\begin{bmatrix} r1 & r3 & r5 & r8 & r12 & r16 & r32 & r13 \\ r2 & r6 & r10 & r19 & r21 & r29 & r44 & r27 \\ r4 & r9 & r18 & r23 & r64 & r37 & r63 & r35 \\ r7 & r17 & r22 & r30 & r36 & r42 & r54 & r39 \\ r11 & r20 & r60 & r34 & r24 & r53 & r45 & r47 \\ r15 & r26 & r33 & r41 & r49 & r55 & r61 & r51 \\ r28 & r38 & r62 & r52 & r43 & r59 & r58 & r56 \\ r14 & r25 & r31 & r40 & r46 & r50 & r57 & r48 \end{bmatrix}$ | $\begin{bmatrix} 16 & 12 & 12 & 14 & 17 & 40 & 51 & 16 \\ 11 & 10 & 14 & 22 & 24 & 61 & 56 & 58 \\ 14 & 13 & 24 & 35 & 99 & 92 & 103 & 64 \\ 16 & 26 & 49 & 60 & 72 & 69 & 62 & 65 \\ 18 & 22 & 120 & 55 & 37 & 103 & 80 & 81 \\ 24 & 40 & 56 & 87 & 95 & 77 & 92 & 103 \\ 51 & 78 & 101 & 104 & 55 & 100 & 112 & 113 \\ 19 & 29 & 57 & 68 & 109 & 98 & 121 & 87 \end{bmatrix}$ |
| $BAS_{11a0}$ [49] | $\begin{bmatrix} r1 & r3 & r5 & r7 & r14 & r17 & r13 & r19 \\ r2 & r8 & r10 & r16 & r28 & r31 & r32 & r30 \\ r4 & r9 & r21 & r23 & r34 & r41 & r40 & r37 \\ r6 & r15 & r22 & r24 & r42 & r45 & r48 & r44 \\ r12 & r25 & r33 & r39 & r49 & r55 & r52 & r51 \\ r20 & r29 & r36 & r47 & r54 & r64 & r63 & r59 \\ r11 & r27 & r38 & r46 & r53 & r62 & r60 & r58 \\ r18 & r26 & r35 & r43 & r50 & r61 & r57 & r56 \end{bmatrix}$ | $\begin{bmatrix} 16 & 12 & 12 & 16 & 19 & 26 & 16 & 22 \\ 11 & 14 & 14 & 40 & 51 & 57 & 51 & 60 \\ 14 & 13 & 24 & 35 & 55 & 87 & 68 & 92 \\ 10 & 24 & 49 & 37 & 69 & 80 & 87 & 56 \\ 17 & 29 & 56 & 65 & 95 & 77 & 104 & 103 \\ 22 & 61 & 72 & 81 & 62 & 99 & 103 & 100 \\ 18 & 58 & 78 & 109 & 103 & 101 & 120 & 112 \\ 24 & 40 & 64 & 55 & 98 & 92 & 121 & 113 \end{bmatrix}$ |
| $BAS_8$ [47] | $\begin{bmatrix} r1 & r3 & r5 & r7 & r14 & r13 & r25 & r18 \\ r2 & r8 & r10 & r17 & r26 & r32 & r45 & r30 \\ r4 & r9 & r15 & r21 & r31 & r36 & r47 & r35 \\ r6 & r16 & r20 & r23 & r38 & r42 & r57 & r39 \\ r12 & r24 & r29 & r37 & r44 & r50 & r59 & r51 \\ r11 & r27 & r34 & r41 & r49 & r56 & r61 & r54 \\ r22 & r43 & r46 & r53 & r58 & r63 & r64 & r62 \\ r19 & r28 & r33 & r40 & r48 & r55 & r60 & r52 \end{bmatrix}$ | $\begin{bmatrix} 16 & 12 & 12 & 16 & 19 & 16 & 29 & 24 \\ 11 & 14 & 14 & 26 & 40 & 51 & 80 & 60 \\ 14 & 13 & 24 & 24 & 57 & 72 & 81 & 64 \\ 10 & 40 & 22 & 35 & 78 & 69 & 121 & 65 \\ 17 & 37 & 61 & 92 & 56 & 98 & 100 & 103 \\ 18 & 58 & 55 & 87 & 95 & 113 & 92 & 62 \\ 49 & 55 & 109 & 103 & 112 & 103 & 99 & 101 \\ 22 & 51 & 56 & 68 & 87 & 77 & 120 & 104 \end{bmatrix}$ |

characteristics, our approach ensures that the Huffman coding table remains unchanged. This is because Huffman coding assigns shorter codes to symbols with higher frequencies, which aligns with our method's strategy.

## 6.3 Performance Assessment

The Cg is a metric commonly used in the literature to evaluate the effectiveness of a compression technique. It measures the level of energy compaction achieved by the transformation process, with a higher Cg indicating better compression efficiency [59]. However, it is important to note that the traditional Cg metric only considers the transformation kernel and does not account for the quantization process. To properly evaluate the performance of the proposed quantization table, it is necessary to use a mCg that takes into account the quantization table [58]. The results of this evaluation can be found in Table 6.2, where the mCg is calculated using the following formula [58]:

$$mR_y = (C \cdot R_x \cdot C^T) \oslash Q, \tag{6.1}$$

where $C$ is a transformation matrix, $\oslash$ is the element-wise division, $R_x$ denotes the covariance matrix of the signal $x$, with its elements based on the exponentiated absolute difference of their corresponding indices, i.e., $\rho^{|i-j|}$. Here, the indices $i$ and $j$ range from 1 to 8. In this study, the correlation factor $\rho$ is set to 0.95, which has been demonstrated to be a reliable approximation for natural images in previous literature [59], $Q$ is defined as follows:

$$Q = \begin{cases} Q_0 & \text{if } QF = 50 \\ round((Q_0 \cdot SF + 50) \div 100) & \text{otherwise ,} \end{cases}$$

where $Q_0$ is one of the quantization tables listed in Table 6.1, used in the compression process, and $SF$ is defined as follows:

$$SF = \begin{cases} 5000 \div QF & \text{if } QF < 50 \\ 200 - 2 \times QF & \text{if } QF > 50, \end{cases}$$

with $QF$ is the quality factor selected by the user to control image quality

Table 6.2: mCg of approximate transforms with the new quantization tables.

| Method | $BAS_{11a0}$ | $BAS_8$ | MCB | $T_p$ | $CB$ | $O_{15}$ | $T_{p1}$ |
|---|---|---|---|---|---|---|---|
| Standard Q | 11.3990 | 11.6158 | 10.7077 | 10.73 | 11.6940 | 13.0094 | 11.4924 |
| Proposed Q | 11.8867 | 11.7336 | 11.0032 | 11.0032 | 12.2190 | 13.8063 | 11.6699 |

and bit-rate. The mCg is computed using the same formula as the conventional Cg, which is shown in equation (6.2). However, it is evaluated in the quantized domain $mR_y$ of $R_x$, which is defined in equation (6.1).

$$mC_g = 10log_{10}\frac{\frac{1}{N}\sum_{i=0}^{N-1}\sigma_{y_i}^2}{\left(\prod_{i=0}^{N-1}\sigma_{y_i}^2||f_i||^2\right)^{\frac{1}{N}}}, \tag{6.2}$$

where $N$ is the number of transform coefficients, $\sigma_{y_i}^2$ is the variance of $i^{th}$ transform coefficient being the $i^{th}$ diagonal element of the matrix $mR_y$ and $||f_i||$ is the 2-norm of the $i^{th}$ basis function of the transform matrix.

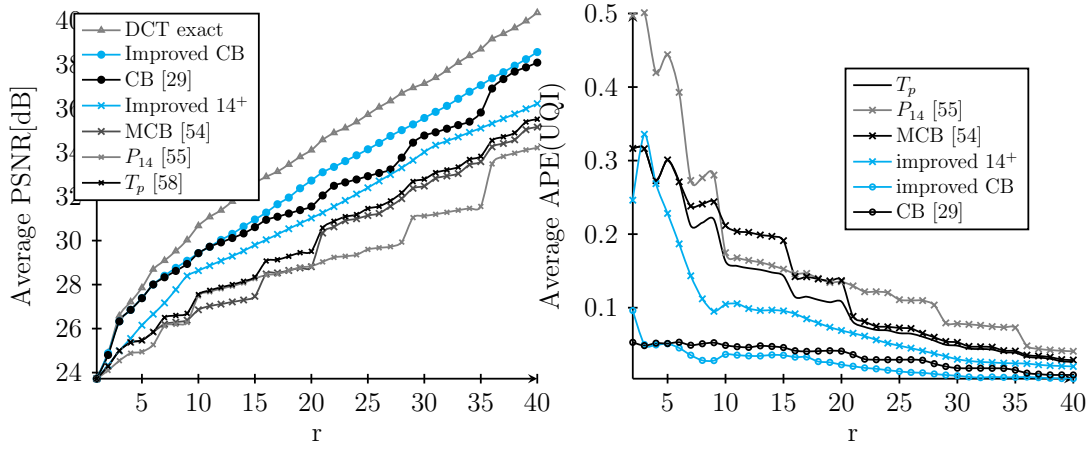
The performance evaluation presented in Table 6.2 demonstrates the superiority of the proposed quantization tables for each approximation compared to the standard quantization table. Specifically, the results indicate a higher $mCg$ value for all approximations when the proposed quantization table is utilized. This observation highlights the effectiveness of the proposed quantization table in achieving improved image compression outcomes.

## 6.4 Application in JPEG-Like Image Compression

The proposed algorithm has been integrated into a JPEG-like image compression framework, utilizing the matrices presented in previous studies [29, 54, 55, 86, 94, 32, 95, 47, 49]. The algorithm was evaluated using a dataset of 47 8-bit images obtained from a public image bank [66]. Each image was divided into $8 \times 8$ blocks and subjected to a 2-D transformation using the matrices specified in [29, 54, 55, 86, 94, 32, 95, 47, 49]. During the compression, only the first $r$ coefficients were retained for each block, with the remaining coefficients being ignored. The value of $r$ varied from 1 to 40.

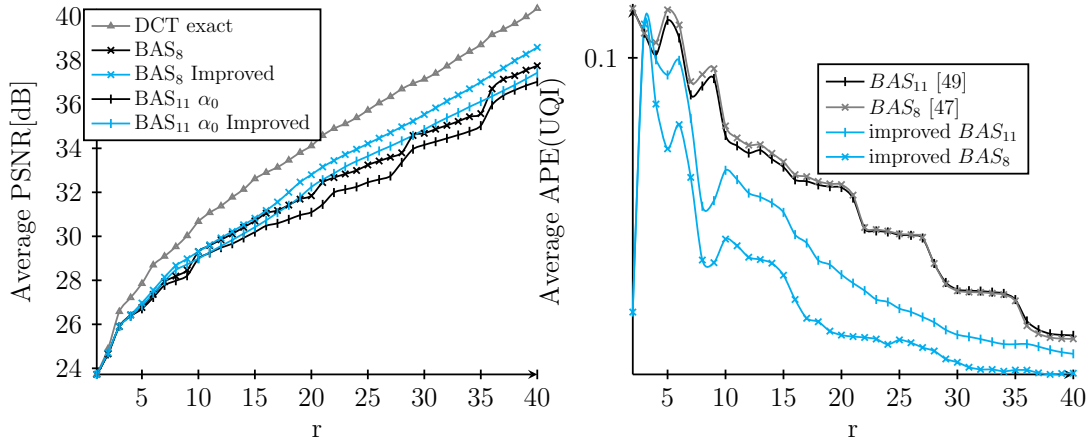
The compressed images were obtained by applying the inverse 2-D transform using the approximation. The evaluation of image degradation was performed by comparing the compressed images with their corresponding original images. To assess the quality of the compressed images, two commonly used measures were employed: the PSNR [61] and the APE of the approximation UQI compared to the UQI of the exact DCT or DTT (APE(UQI)) [62].

These measures were applied to the 47 images for each value of $r$ and the average image quality measures were calculated. The proposed algorithm has been integrated into the JPEG-Like compression framework to assess the improvement in quality. The results of the improved algorithm are presented in Fig. 6.4 a. The plots in cyan represent the improved results.



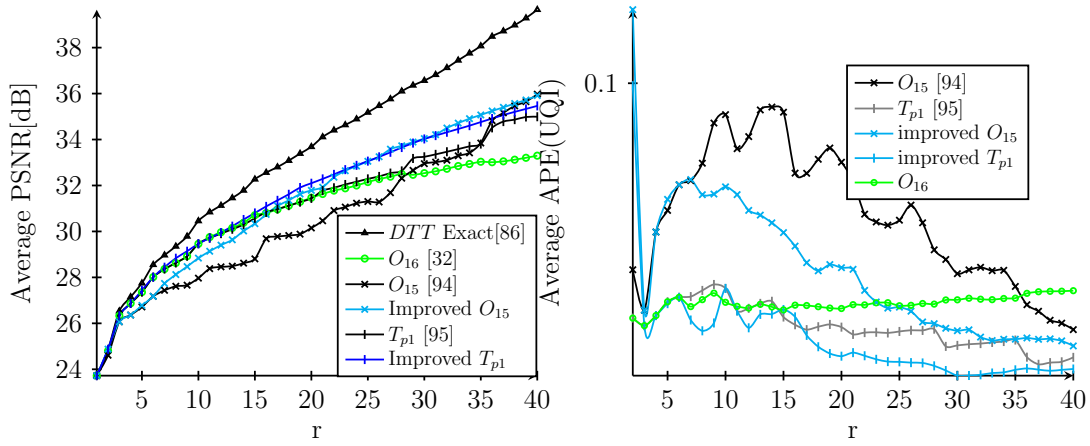

(a) Average PSNR for DCT approximations

(b) Average APE(UQI) for DCT approximations.




(c) Average PSNR for BAS series DCT approximations

(d) Average APE(UQI) for BAS series DCT approximations.




(e) Average PSNR for DTT approximations

(f) Average APE(UQI) for DTT approximations.

Figure 6.4: Quality evaluation vs the number of retained coefficients (r).

### 6.4.1 Results and Discussion of DCT Approximations

The results depicted in Fig. 6.4 demonstrate that the proposed algorithm significantly outperforms the original CB approximation [29]. In terms of PSNR, the average gain in quality was found to be 0.5687 dB. On the other hand, the MCB [54], $\mathbf{P_{14}}$ [55] and $T_p$ [58] approximations show a significant increase in quality when compared to CB. Particularly, the $\mathbf{P_{14}}$ approximation demonstrates an increase of almost 2 dB when the 36th coefficient is included. When the proposed algorithm is applied to the aforementioned approximations within the JPEG-Like channel, significant improvements in quality are observed. Specifically, the algorithm demonstrates an average improvement of nearly 1 dB for the MCB approximation [54], exceeding 2 dB for the $\mathbf{P_{14}}$ approximation [55], and approximately 1 dB for the $T_p$ approximation [58]. Furthermore, when evaluating the performance of the proposed transformation using APE(UQI), the results demonstrate lower errors compared to the original transformations. Specifically, the improved $14^+$ transformation exhibits a significant reduction of approximately 10% in errors when the 9th coefficient is added. These improvements highlight the superior performance of the proposed algorithm in enhancing the quality of compressed images.

### 6.4.2 Results and Discussion of BAS Approximations

The proposed algorithm was also tested on BAS series approximations to further evaluate its effectiveness. Specifically, two well-known approximations, $BAS_8$ [47] and $BAS_{11}$ [49], with $\alpha = 0$ were selected for testing. The results demonstrate that the proposed approximation yields better image quality when compared to the original approximations. When implemented on $BAS_8$ [47], the proposed algorithm yields a gain in quality of approximately 1 dB for $19 < r < 35$ compared to the standard order. Furthermore, it outperforms the approximation of $BAS_{11}$ [49] without the proposed algorithm, despite the latter using higher arithmetic complexity. When the proposed algorithm was applied to $BAS_{11}$ [49], it yielded an important gain in quality that exceeded 1 dB, allowing it to lead in terms of image quality. The results are presented in Fig. 6.4 b.

### 6.4.3 Results and Discussion of DTT Approximations

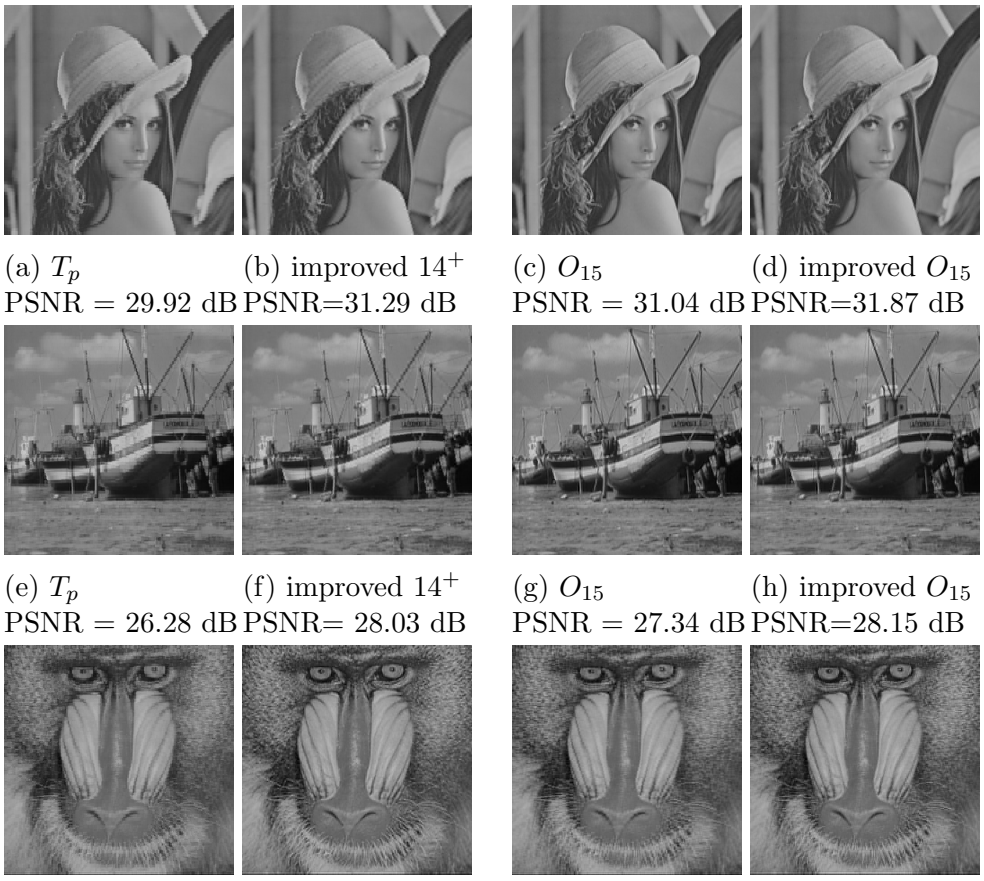
The algorithm has been applied to the transformations $O_{15}$ and $T_{p1}$, and the results are presented in Fig. 6.4 c. The results show an important improvement in terms of quality, where the proposed algorithm allows the

transformation to behave better, with a gain that reaches more than 1.9 dB in some cases for $O_{15}$. The average gain in quality of the proposed method is equal to 0.9717 dB and 0.4355 dB for $O_{15}$ [94] and $T_{p1}$ [95], respectively. The approximation $O_{16}$ outperforms the approximation $O_{15}$ without the proposed algorithm with an important gain in quality, especially before the intercept point. The original intercept point was at $r = 29$ without the proposed algorithm. However, with the addition of the proposed algorithm, the intercept point returns to $r = 17$. This change also leads to a decrease in quality gains for the $O_{16}$ approximation compared to $O_{15}$ with our method. Moreover, a higher gain after the intercept point. It does not mean that the improved $O_{15}$ is better than $O_{16}$, because the approximation $O_{16}$ uses the matrix and its transpose, which allows to use a single architecture. In contrast, the approximation $O_{15}$ uses the matrix and its inverse. Therefore, two architectures are required.

### 6.4.4 Visual Evaluation

A visual evaluation was carried out by selecting three image types (as seen in Fig. 6.5): 'Lena' (with low-frequency components), 'Boat' (with medium-frequency features), and 'Baboon' (with high-frequency elements). This evaluation involved subjecting these images to JPEG-like compression while retaining 10 and 19 coefficients. The results of the evaluation are presented in Fig. 6.5 and Table 6.3.

The effectiveness of the proposed method for image compression was evaluated by visual assessment of the reconstructed images. The evaluation was performed on a set of test images using two different numbers of retained coefficients, 9 and 19. The visual evaluation of the reconstructed images was carried out using PSNR. Fig. 6.5 presents the visual evaluation of the reconstructed images using the proposed algorithm with 9 retained coefficients, respectively. The results demonstrate a significant improvement in image quality for all the test images compared to the original images. The improvement is more than 0.3 dB in most cases, with the improved $O_{15}$ for 19 retained coefficients showing a particularly noticeable improvement in the 'Boat' image (Table 6.3) with an increase of over 1.3 dB. It is noteworthy that the proposed algorithm achieved these results without requiring any additional arithmetic complexity. These results confirm the effectiveness of the proposed method in improving the quality of transform-based image compression.



(a) $T_p$ PSNR = 29.92 dB (b) improved $14^+$ PSNR=31.29 dB (c) $O_{15}$ PSNR = 31.04 dB (d) improved $O_{15}$ PSNR=31.87 dB

(e) $T_p$ PSNR = 26.28 dB (f) improved $14^+$ PSNR= 28.03 dB (g) $O_{15}$ PSNR = 27.34 dB (h) improved $O_{15}$ PSNR=28.15 dB

(i) $T_p$ PSNR = 22.7 dB (j) $O_{15}$ PSNR = 22.67 dB (k) improved $14^+$ PSNR= 23.20 dB (l) improved $O_{15}$ PSNR=23.07 dB

Figure 6.5: Reconstructed images with 9 retained coefficients ($r = 9$)

Table 6.3: Quality of reconstructed images (PSNR[$dB$]) with 19 retained coefficients.

| Method | Lena | Boat | Baboon |
|---|---|---|---|
| $T_p$ [58] | 32.9 | 29.15 | 24.78 |
| Improved $14^+$ | 33.12 | 30.35 | 24.7 |
| $O_{15}$ [94] | 33.43 | 29.86 | 23.99 |
| Improved $O_{15}$ | 34.55 | 31.58 | 24.85 |

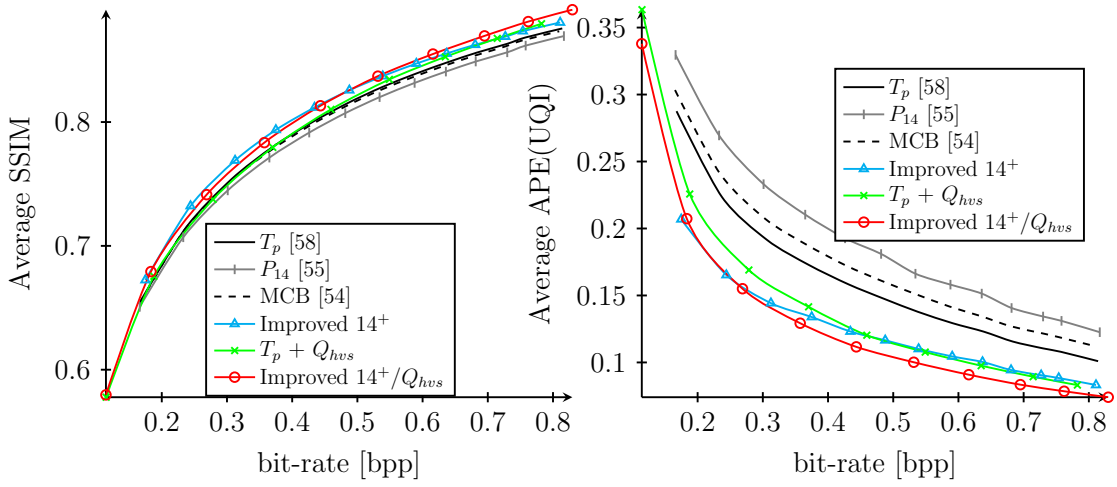
## 6.5 Application in JPEG Image Compression

The proposed algorithm for image compression was developed with the consideration that it was trained using JPEG-LIKE compression and did not account for the quantization process. However, the quantization tables and coefficients order for each approximation were established based on the algorithm. To evaluate the effectiveness of the proposed quantization tables and coefficients order, we assess their performance in the JPEG compression chain. This chain includes the quantization step along with the Huffman coding to ensure that the proposed approach has no negative impact on the compression efficiency.

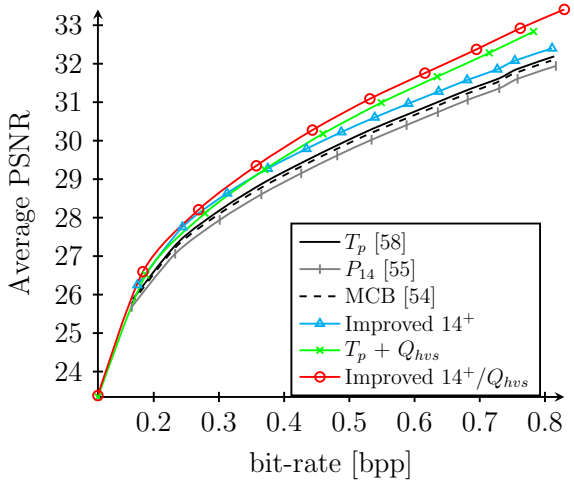
To evaluate the performance, the same database [66] used in the optimization was used for the evaluation using JPEG-Like compression. This database [66], contains a variety of images that simulate different scenarios. This ensures that the results are applicable to a wide range of image types. However, for added validation and confidence, 24 images from the Kodak Photo CD dataset [128] were also chosen. Moreover, comparable results can be obtained for other types of compression that involve a quantization process. These images were compressed using the JPEG standard with the $T_p$ [95] approximation. Therefore, the quality of the reconstructed images was assessed using established metrics, including SSIM [67], PSNR [61], and APE(UQI) [62]. The results of this evaluation are depicted in Fig. 6.6.

The evaluation of the JPEG channel highlights the significant improvements in image quality achieved by the proposed method compared to the standard quantization and zig-zag approaches. Results in Fig. 6.6 clearly demonstrate a substantial gain in quality across all evaluated metrics. Notably, all DCT approximations with 14 additions yield similar results, with variations in the quantization tables and coefficients order.

In terms of PSNR, the proposed method exhibits an average gain in quality of approximately 0.3 dB compared to the $T_p$ approximation. This improvement indicates the effectiveness of the proposed quantization tables and coefficients order in enhancing the image quality. It is important to




(a) Average SSIM vs bit-rate (b) Average APE(UQI) vs bit-rate.




(c) Average PSNR vs bit-rate.

Figure 6.6: Evaluation of image quality measures for the images in the dataset [128] using 14 additions DCT approximations at different compression ratios (bit-rates).

highlight that these gains are achieved without any increase in arithmetic complexity, making the proposed method a highly efficient solution for image compression.

To comprehensively assess our algorithm, we conducted experiments with an alternative quantization table based on the human visual system ($Q_{hvs}$) [121]. The table was restructured using the order of the coefficients generated by our algorithm, resulting in the modified quantization table presented in Table 6.5, alongside the original version [121].

The outcomes of implementing this modified quantization table are depicted in Fig. 6.6. Notably, our proposed method demonstrates a substantial improvement in image quality across all metrics, with a particular emphasis on PSNR. The enhanced $Q_{hvs}$ table, specifically tailored for $T_p$, denoted as (improved $14^{+}/Q_{hvs}$), surpasses all other quantization tables in terms of PSNR. This underlines the versatility of our method, showing its applicability across various image types, as validated on two distinct databases [66, 128].

Furthermore, our approach is compatible with a range of quantization tables proposed in the literature, as evidenced by successful testing with two different quantization tables: the standard table and $Q_{hvs}$ [121].

Table 6.4: Improved quantization table for $T_p$ based on $Q_{hvs}$

| | |
|---|---|
| Original $Q_{hvs}$ | $\begin{pmatrix} 16 & 16 & 16 & 16 & 17 & 18 & 21 & 24 \\ 16 & 16 & 16 & 16 & 17 & 19 & 22 & 25 \\ 16 & 16 & 17 & 18 & 20 & 22 & 25 & 29 \\ 16 & 16 & 18 & 21 & 24 & 27 & 31 & 36 \\ 17 & 17 & 20 & 24 & 30 & 35 & 41 & 47 \\ 18 & 19 & 22 & 27 & 35 & 44 & 54 & 65 \\ 21 & 22 & 25 & 31 & 41 & 54 & 70 & 88 \\ 24 & 25 & 29 & 36 & 47 & 65 & 88 & 115 \end{pmatrix}$ |
| Improved $Q_{hvs}$ | $\begin{pmatrix} 16 & 16 & 16 & 16 & 24 & 21 & 22 & 27 \\ 16 & 16 & 18 & 24 & 16 & 22 & 27 & 24 \\ 16 & 18 & 21 & 16 & 19 & 44 & 25 & 22 \\ 17 & 16 & 16 & 19 & 20 & 25 & 35 & 65 \\ 17 & 16 & 18 & 17 & 29 & 25 & 54 & 65 \\ 17 & 18 & 20 & 31 & 31 & 54 & 30 & 70 \\ 21 & 24 & 25 & 29 & 36 & 41 & 47 & 88 \\ 22 & 17 & 35 & 36 & 41 & 47 & 88 & 115 \end{pmatrix}$ |

JPEG compression was also applied to the 'Lena,' 'Boat,' and 'Baboon' images. The resulting images can be found in Fig. 6.7 and Table 6.5. It was observed that the proposed quantization tables and coefficients order produced better results in terms of quality when applied to the approximations $T_p$ [95] and $O_{15}$ [94]. In comparison to the standard quantization tables and coefficients order, the gain in quality was approximately 0.5 dB in certain cases, such as the 'Lena' image at 0.4 *bpp* and the 'Boat' image when compressed using $O_{15}$. The average gain in quality was approximately 0.3 dB. The results suggest that the proposed algorithm can potentially improve the quality of compressed images without the need for additional arithmetic

Table 6.5: Quality of reconstructed images (PSNR[$dB$]) at a bit-rate around 0.8 bpp

| Method | Lena | Boat | Baboon |
|---|---|---|---|
| MCB [54] | 35.80 | 31.80 | 25.23 |
| $\mathbf{P_{14}}$ [55] | 35.49 | 31.62 | 25.06 |
| $T_p$ [58] | 35.97 | 32.08 | 25.27 |
| Improved $14^+$ | 36.28 | 32.36 | 25.41 |
| $O_{15}$ [94] | 35.06 | 30.89 | 24.03 |
| Improved $O_{15}$ | 35.40 | 31.23 | 24.50 |
| $T_{p1}$ [95] | 36.15 | 32.18 | 25.61 |
| Improved $T_{p1}$ | 36.20 | 32.16 | 25.68 |

complexity. The effectiveness of the proposed quantization tables and coefficients order further enhances the performance of the algorithm when used in the JPEG compression chain.

## 6.6 Conclusion

This chapter presents an innovative approach to enhance image compression efficiency for various DTT and DCT approximations. It highlights the importance of customized quantization tables and coefficient orders tailored for each approximation. Importantly, our implementation does not introduce additional computational complexity, thanks to an algorithm that optimizes coefficient order based on their impact. We applied this algorithm to various approximations, such as MCB, CB, BAS series, $\mathbf{P_{14}}$, and $O_{15}$ [94, 54, 55, 29, 47, 49], resulting in an average quality improvement of over 0.3 dB. Visual evaluations conducted using both JPEG-LIKE and standard JPEG compression confirm the effectiveness of our approach.

This work offers a practical solution for image compression and paves the way for testing the method in video coding standards. Importantly, our approach is not limited to specific approximations and can be applied to new ones to enhance their performance. Our work has immediate relevance in the context of wireless networks where efficient image compression plays a critical role in minimizing bandwidth consumption, ensuring faster transmission, and ultimately enhancing the overall user experience. Future research directions could involve defining entirely new quantization tables tailored for each approximation, as well as exploring further improvements and applications across various contexts.

(a) $T_p$
bit-rate=0.41 *bpp*
PSNR = 32.68 dB

(b) improved $14^+$
bit-rate=0.4 *bpp*
PSNR=32.94 dB

(c) $O_{15}$
bit-rate=0.4 *bpp*
PSNR = 32.29 dB

(d) improved $O_{15}$
bit-rate=0.4 *bpp*
PSNR=32.58 dB

(e) $T_p$
bit-rate=0.79 *bpp*
PSNR = 35.97 dB

(f) improved $14^+$
bit-rate=0.79 *bpp*
PSNR= 36.28 dB

(g) $O_{15}$
bit-rate=0.81 *bpp*
PSNR = 35.06 dB

(h) improved $O_{15}$
bit-rate=0.79 *bpp*
PSNR=35.40 dB

Figure 6.7: Reconstructed images for bit-rate $\in \{0.4, 0.8\}$

# General Conclusion

This thesis has explored and addressed several key challenges in the design and implementation of efficient embedded monitoring systems, particularly in the context of WSNs and IoT applications. The focus has been on developing low-complexity compression techniques and efficient coding methods that reduce the energy consumption and computational complexity of sensor nodes, all while maintaining an acceptable level of image quality.

The rapid growth of IoT and its integration with WSNs has brought forward the need for more efficient resource management in embedded systems, where energy, bandwidth, and storage are limited. Embedded monitoring applications require continuous data acquisition and real-time analysis, making it crucial to optimize data transmission and reduce computational demands on the sensor nodes. In this context, this research focused on the development and evaluation of new low-complexity DCT and DTT approximations, as well as methodologies for optimizing image compression.

A comparative study of existing techniques was carried out, leading to the proposal of novel low-complexity transform approximations tailored for resource-constrained environments. These transforms were evaluated based on their performance in terms of PSNR, SSIM, and bitrate, and compared against existing methods to demonstrate their effectiveness. Moreover, the thesis explored how these techniques can be applied in real-world systems by implementing them on FPGA, which validated their feasibility in practical embedded monitoring applications.

One of the significant outcomes of this research is the development of a pruned version of the proposed DCT approximation, which further reduces the computational cost without compromising the overall performance of the system. This enables the design of energy-efficient systems that are highly suitable for real-time monitoring in environments such as industrial settings, wildlife observation, or environmental monitoring.

# Main Contributions

The primary contributions of this thesis can be summarized as follows:

- **Low-complexity Transform Approximation:** The development of novel DCT and DTT approximations that significantly reduce the number of arithmetic operations required, making them ideal for embedded monitoring applications where energy efficiency is crucial. These techniques showed improvements in image compression performance while maintaining image quality compared to existing methods.

- **Pruned Transform Versions:** The introduction of pruned versions of the transform approximations further reduces the computational complexity while maintaining an acceptable image quality. This allows for more efficient data transmission, particularly for ROI in image-based monitoring applications.

- **FPGA-based Validation:** The proposed techniques were implemented on FPGA, demonstrating the practicality and efficiency of the low-complexity transformations in real-world systems. The validation results confirm that the proposed methods can be integrated into embedded monitoring systems with limited hardware resources, providing both energy savings and performance improvements.

- **Optimized Quantization and Coding:** The thesis also contributed by developing an optimized methodology for generating quantization tables and ordering coefficients, which further enhanced the image quality while minimizing the compression complexity.

# Limitations of the Work

While the contributions of this research provide significant improvements in the area of embedded monitoring, several limitations should be acknowledged. First, the proposed techniques were tested primarily in simulation environments and implemented on specific hardware platforms (FPGA), which may limit their generalizability to other hardware architectures or more complex sensor networks.

Additionally, the proposed algorithms primarily focused on reducing the arithmetic complexity of image compression. Although the proposed transformations achieved the lowest number of operations (14 additions) while still delivering superior image quality compared to their competitors, further reducing the number of operations in the transformation step seems unlikely. The fundamental constraints of the transformation process, particularly for efficient low-complexity kernels, may limit additional reductions in

arithmetic complexity. However, future work could explore enhancing image quality by optimizing the quantization process or developing more efficient quantization tables, which could improve overall compression performance without increasing transformation complexity.

In general, the transformation step is not inherently lossy in most traditional approximations, particularly in orthogonal transformations. However, DTT approximations, which are near-orthogonal, introduce a degree of loss during the transformation process. This means that valuable information is discarded even before the quantization stage, which further degrades image quality. Moreover, near-orthogonal approximations require distinct architectures for both forward and inverse transformations, leading to increased hardware resource consumption.

Future work could focus on reducing the errors introduced during the transformation step for near-orthogonal kernels, potentially improving both accuracy and efficiency. Furthermore, designing a unified hardware architecture for both forward and inverse transformations could significantly reduce hardware resource requirements, making these approximations more feasible for real-world embedded applications.

Finally, while the pruned transform approximations offer significant reductions in computational complexity, there may be further room for improvement, especially in highly constrained environments where even minor reductions in resource consumption can have a large impact on system performance.

# Future Work

Building on the findings of this research, several avenues for future exploration present themselves. These include:

- **Enhanced Hardware Integration:** Although the proposed techniques were successfully implemented on FPGA, future work could focus on integrating them with more advanced hardware platforms or low-power processors, such as microcontroller units (MCUs) or system-on-chip (SoC) designs, to further optimize the energy efficiency of embedded systems.

- **Further Compression Optimization:** The proposed methodologies for quantization and coding could be further optimized, particularly with respect to specific applications where even greater efficiency in data compression and transmission is required. Investigating new ways to adaptively adjust compression parameters based on real-time data conditions could also be a valuable direction.

- **Real-world Deployment:** Finally, future work could involve the de-

ployment of the proposed techniques in larger-scale, real-world WSNs and IoT deployments. This would provide further insights into the practical challenges of implementing low-complexity compression techniques in diverse and dynamic environments, and allow for the validation of the techniques in live monitoring scenarios.

# Final Remarks

The work presented in this thesis has advanced the state-of-the-art in embedded monitoring systems by providing new, efficient methodologies for data compression and coding within resource-constrained WSNs environments. The proposed techniques, validated through simulation and FPGA implementation, offer practical solutions for improving energy efficiency and maintaining data quality in embedded systems. This research opens up new possibilities for the development of smarter, more efficient, and more sustainable monitoring systems, which are critical for the growing field of IoT and its many applications.

# Bibliography


[1] L. Belkhir and A. Elmeligi, "Assessing ict global emissions footprint: Trends to 2040 & recommendations," *Journal of cleaner production*, vol. 177, pp. 448–463, 2018.

[2] M. M. Baig, H. GholamHosseini, and M. J. Connolly, "Mobile healthcare applications: System design review, critical issues and challenges," *Australasian physical & engineering sciences in medicine*, vol. 38, pp. 23–38, 2015.

[3] L. I. Minchala, J. Peralta, P. Mata-Quevedo, and J. Rojas, "An approach to industrial automation based on low-cost embedded platforms and open software," *Applied Sciences*, vol. 10, no. 14, p. 4696, 2020.

[4] N. Vidakis, M. A. Lasithiotakis, and E. Karapidakis, "Environmental monitoring through embedded system and sensors," in *2017 52nd International Universities Power Engineering Conference (UPEC)*, IEEE, 2017, pp. 1–7.

[5] J. Lee, H.-A. Kao, and S. Yang, "Service innovation and smart analytics for industry 4.0 and big data environment," *Procedia cirp*, vol. 16, pp. 3–8, 2014.

[6] W. Kassab and K. A. Darabkh, "A–z survey of internet of things: Architectures, protocols, applications, recent advances, future directions and recommendations," *Journal of Network and Computer Applications*, vol. 163, p. 102 663, 2020.

[7] J. L. Hennessy and D. A. Patterson, *Computer architecture: a quantitative approach*. Morgan kaufmann, 2017.

[8] W. Aspray, "The intel 4004 microprocessor: What constituted invention?" *IEEE Annals of the History of Computing*, vol. 19, no. 3, pp. 4–15, 1997.

[9] T. Noergaard, *Embedded systems architecture: a comprehensive guide for engineers and programmers*. Newnes, 2012.

[10] Kamal, *Embedded Systems: Architechture, Programming and Design*. McGraw-Hill Science/Engineering/Math, 2006.

[11] P. Marwedel, *Embedded system design: embedded systems foundations of cyber-physical systems, and the internet of things*. Springer Nature, 2021.

[12] Z. Zhang and J. Li, "A review of artificial intelligence in embedded systems," *Micromachines*, vol. 14, no. 5, p. 897, 2023.

[13] J.-W. Kim, H.-W. Choi, S.-K. Kim, and W. S. Na, "Review of image-processing-based technology for structural health monitoring of civil infrastructures," *Journal of Imaging*, vol. 10, no. 4, p. 93, 2024.

[14] C. R. Farrar and K. Worden, "An introduction to structural health monitoring," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, pp. 303–315, 2007.

[15] K. Mostafa and T. Hegazy, "Review of image-based analysis and applications in construction," *Automation in Construction*, vol. 122, p. 103 516, 2021.

[16] L. Parente *et al.*, "Image-based monitoring of cracks: Effectiveness analysis of an open-source machine learning-assisted procedure," *Journal of Imaging*, vol. 8, no. 2, p. 22, 2022.

[17] E. Perakakis, G. Mastorakis, and I. Kopanakis, "Social media monitoring: An innovative intelligent approach," *Designs*, vol. 3, no. 2, p. 24, 2019.

[18] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[19] R. Ananthanarayanan *et al.*, "Photon: Fault-tolerant and scalable joining of continuous data streams," in *Proceedings of the 2013 ACM SIGMOD international conference on management of data*, 2013, pp. 577–588.

[20] S. Mittal, "A survey of techniques for improving energy efficiency in embedded computing systems," *International Journal of Computer Aided Engineering and Technology*, vol. 6, no. 4, pp. 440–459, 2014.

[21] F. Chen, D. A. Koufaty, and X. Zhang, "Understanding intrinsic characteristics and system implications of flash memory based solid state drives," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 1, pp. 181–192, 2009.

[22] S. T. Ahmed and S. Sankar, "Investigative protocol design of layer optimized image compression in telemedicine environment," *Procedia Computer Science*, vol. 167, pp. 2617–2622, 2020.

[23] H. Huang, G. Coatrieux, H. Shu, L. Luo, and C. Roux, "Blind integrity verification of medical images," *IEEE transactions on information technology in biomedicine*, vol. 16, no. 6, pp. 1122–1126, 2012.

[24] G. Cheng, X. Xie, J. Han, L. Guo, and G.-S. Xia, "Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 3735–3756, 2020.

[25] G. F. Shidik, E. Noersasongko, A. Nugraha, P. N. Andono, J. Jumanto, and E. J. Kusuma, "A systematic review of intelligence video surveillance: Trends, techniques, frameworks, and datasets," *IEEE Access*, vol. 7, pp. 170 457–170 473, 2019.

[26] A. Boulmaiz, N. Doghmane, S. Harize, N. Kouadria, and D. Messadeg, "The use of wsn (wireless sensor network) in the surveillance of endangered bird species," in *Advances in ubiquitous computing*, Elsevier, 2020, pp. 261–306.

[27] R. K. Senapati, U. C. Pati, and K. K. Mahapatra, "Reduced memory, low complexity embedded image compression algorithm using hierarchical listless discrete tchebichef transform," *IET Image Processing*, vol. 8, no. 4, pp. 213–238, 2014.

[28] S.-W. Lee and H.-Y. Kim, "An energy-efficient low-memory image compression system for multimedia iot products," *EURASIP Journal on Image and Video Processing*, vol. 2018, pp. 1–15, 2018.

[29] R. J. Cintra and F. M. Bayer, "A dct approximation for image compression," *IEEE Signal Processing Letters*, vol. 18, no. 10, pp. 579–582, 2011.

[30] R. J. Cintra, F. M. Bayer, and C. Tablada, "Low-complexity 8-point dct approximations based on integer functions," *Signal Processing*, vol. 99, pp. 201–214, 2014.

[31] R. S. Oliveira, R. J. Cintra, F. M. Bayer, T. L. da Silveira, A. Madanayake, and A. Leite, "Low-complexity 8-point dct approximation based on angle similarity for image and video coding," *Multidimensional Systems and Signal Processing*, vol. 30, pp. 1363–1394, 2019.

[32] P. A. Oliveira, R. J. Cintra, F. M. Bayer, S. Kulasekera, and A. Madanayake, "Low-complexity image and video coding based on an approximate discrete tchebichef transform," *IEEE transactions on circuits and systems for video technology*, vol. 27, no. 5, pp. 1066–1076, 2016.

[33] V. A. Coutinho, R. J. Cintra, F. M. Bayer, S. Kulasekera, and A. Madanayake, "A multiplierless pruned dct-like transformation for image and video compression that requires ten additions only," *Journal of Real-Time Image Processing*, vol. 12, pp. 247–255, 2016.

[34] M. Jridi, A. Alfalou, and P. K. Meher, "A generalized algorithm and reconfigurable architecture for efficient and scalable orthogonal approximation of dct," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 2, pp. 449–457, 2014.

[35] A. Sakhri *et al.*, "Audio-visual low power system for endangered waterbirds monitoring," *IFAC-PapersOnLine*, vol. 55, no. 5, pp. 25–30, 2022.

[36] L. M. Oliveira and J. J. Rodrigues, "Wireless sensor networks: A survey on environmental monitoring.," *J. Commun.*, vol. 6, no. 2, pp. 143–151, 2011.

[37] R. Monika and S. Dhanalakshmi, "An efficient medical image compression technique for telemedicine systems," *Biomedical Signal Processing and Control*, vol. 80, p. 104 404, 2023.

[38] H. Kasban, S. Nassar, and M. A. El-Bendary, "Medical images transmission over wireless multimedia sensor networks with high data rate," *Analog Integrated Circuits and Signal Processing*, vol. 108, no. 1, pp. 125–140, 2021.

[39] A. Aliouat, N. Kouadria, M. Maimour, S. Harize, and N. Doghmane, "Region-of-interest based video coding strategy for rate/energy-constrained smart surveillance systems using wmsns," *Ad Hoc Networks*, vol. 140, p. 103 076, 2023.

[40] A. Aliouat, N. Kouadria, S. Harize, and M. Maimour, "An efficient low complexity region-of-interest detection for video coding in wireless visual surveillance," *IEEE Access*, 2023.

[41] N. R. Kidwai, E. Khan, and M. Reisslein, "Zm-speck: A fast and memoryless image coder for multimedia sensor networks," *IEEE Sensors Journal*, vol. 16, no. 8, pp. 2575–2587, 2016.

[42] B. K. Mohanty, "Approximate lifting 2-d dwt hardware design for image encoder of wireless visual sensors," *IEEE Sensors Journal*, 2023.

[43] G. K. Wallace, "The jpeg still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[44] S. Harize, A. Mefoued, N. Kouadria, and N. Doghmane, "Hevc transforms with reduced elements bit depth," *Electronics Letters*, vol. 54, no. 22, pp. 1278–1280, 2018.

[45] R. Clark, "Relation between the karhunen-loeve and cosine transform," *Proc. IEEE*, vol. 128, no. 11, pp. 359–360, 1981.

[46] J. Žádník, M. Mäkitalo, J. Vanne, and P. Jääskeläinen, "Image and video coding techniques for ultra-low latency," *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–35, 2022.

[47] S. Bouguezel, M. O. Ahmad, and M. Swamy, "Low-complexity 8× 8 transform for image compression," *Electronics Letters*, vol. 44, no. 21, pp. 1249–1250, 2008.

[48] S. Bouguezel, M. O. Ahmad, and M. Swamy, "A novel transform for image compression," in *2010 53rd IEEE International Midwest Symposium on Circuits and Systems*, IEEE, 2010, pp. 509–512.

[49] S. Bouguezel, M. O. Ahmad, and M. Swamy, "A low-complexity parametric transform for image compression," in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, IEEE, 2011, pp. 2145–2148.

[50] T. L. Da Silveira, D. R. Canterle, D. F. Coelho, V. A. Coutinho, F. M. Bayer, and R. J. Cintra, "A class of low-complexity dct-like transforms for image and video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[51] N. Brahimi, T. Bouden, T. Brahimi, and L. Boubchir, "Lossy image compression based on efficient multiplier-less 8-points dct," *Multimedia Systems*, vol. 28, no. 1, pp. 171–182, 2022.

[52] A. Khalili Sadaghiani and B. Forouzandeh, "Low-power hardware-efficient memory-based dct processor," *Journal of Real-Time Image Processing*, vol. 19, no. 6, pp. 1105–1121, 2022.

[53] K. Mechouek, N. Kouadria, N. Doghmane, and N. Kaddeche, "Low complexity dct approximation for image compression in wireless image sensor networks," *Journal of Circuits, Systems and Computers*, vol. 25, no. 08, p. 1 650 088, 2016.

[54] F. Bayer and R. Cintra, "Dct-like transform for image compression requires 14 additions only," *Electronics Letters*, vol. 48, no. 15, p. 1, 2012.

[55] U. S. Potluri, A. Madanayake, R. J. Cintra, F. M. Bayer, S. Kulasekera, and A. Edirisuriya, "Improved 8-point approximate dct for image and video compression requiring only 14 additions," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 6, pp. 1727–1740, 2014.

[56] S. Bouguezel, M. O. Ahmad, and M. Swamy, "A fast 8× 8 transform for image compression," in *2009 International Conference on Microelectronics-ICM*, IEEE, 2009, pp. 74–77.

[57] C. Araar, S. Ghanemi, M. Benmohammed, and H. Atoui, "Pruned improved eight-point approximate dct for image encoding in visual sensor networks requiring only ten additions," *Journal of real-time image processing*, vol. 17, pp. 1597–1608, 2020.

[58] A. Mefoued, N. Kouadria, S. Harize, and N. Doghmane, "Improving image encoding quality with a low-complexity DCT approximation using 14 additions," *Journal of Real-Time Image Processing*, vol. 20, no. 3, p. 58, 2023, ISSN: 1861-8219.

[59] V. Britanak, P. C. Yip, and K. R. Rao, *Discrete cosine and sine transforms: general properties, fast algorithms and integer approximations*. Elsevier, 2010.

[60] N. J. Higham, "Computing the polar decomposition—with applications," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 4, pp. 1160–1174, 1986.

[61] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008.

[62] Z. Wang, "A universal image quality index," *IEEE signal processing letters*, vol. 9, no. 3, pp. 81–84, 2002.

[63] R. E. Blahut, *Fast algorithms for signal processing*. Cambridge University Press, 2010.

[64] T. I. Haweel, "A new square wave transform based on the dct," *Signal processing*, vol. 81, no. 11, pp. 2309–2319, 2001.

[65] N. Zidani, N. Kouadria, N. Doghmane, and S. Harize, "Low complexity pruned dct approximation for image compression in wireless multimedia sensor networks," in *2019 5th International Conference on Frontiers of Signal Processing (ICFSP)*, IEEE, 2019, pp. 26–30.

[66] U. of Southern California, "The usc-sipi image database ⟨http://sipi.usc.edu/database/⟩," *Signal and Image Processing Institute*, 2011.

[67] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[68] S. Jeong, S. Jeong, S. S. Woo, and J. H. Ko, "An overhead-free region-based jpeg framework for task-driven image compression," *Pattern Recognition Letters*, vol. 165, pp. 1–8, 2023.

[69] S. H. Kim, J. H. Park, and J. H. Ko, "Target-dependent scalable image compression using a reconfigurable recurrent neural network," *IEEE Access*, vol. 9, pp. 119 418–119 429, 2021.

[70] Q. Wang, L. Shen, and Y. Shi, "Recognition-driven compressed image generation using semantic-prior information," *IEEE Signal Processing Letters*, vol. 27, pp. 1150–1154, 2020.

[71] E. Feig and S. Winograd, "Fast algorithms for the discrete cosine transform," *IEEE Transactions on Signal processing*, vol. 40, no. 9, pp. 2174–2193, 1992.

[72] H. Hou, "A fast recursive algorithm for computing the discrete cosine transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 10, pp. 1455–1461, 1987.

[73] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, "Practical fast 1-d dct algorithms with 11 multiplications," in *International Conference on Acoustics, Speech, and Signal Processing,*, IEEE, 1989, pp. 988–991.

[74] M. Jridi and P. K. Meher, "Scalable approximate dct architectures for efficient hevc-compliant video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 8, pp. 1815–1825, 2016.

[75] G. Paim, L. M. G. Rocha, H. Amrouch, E. A. C. da Costa, S. Bampi, and J. Henkel, "A cross-layer gate-level-to-application co-simulation for design space exploration of approximate circuits in hevc video encoders," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 10, pp. 3814–3828, 2019.

[76] M. A. Suhail and M. S. Obaidat, "Digital watermarking-based dct and jpeg model," *IEEE transactions on instrumentation and measurement*, vol. 52, no. 5, pp. 1640–1647, 2003.

[77] C. Das, S. Panigrahi, V. K. Sharma, and K. Mahapatra, "A novel blind robust image watermarking in dct domain using inter-block coefficient correlation," *AEU-International Journal of Electronics and Communications*, vol. 68, no. 3, pp. 244–253, 2014.

[78] B. A. Lungisani, C. K. Lebekwe, A. M. Zungeru, and A. Yahya, "Image compression techniques in wireless sensor networks: A survey and comparison," *IEEE Access*, vol. 10, pp. 82 511–82 530, 2022.

[79] N. Brahimi, T. Bouden, T. Brahimi, and L. Boubchir, "A novel and efficient 8-point dct approximation for image compression," *Multimedia Tools and Applications*, vol. 79, no. 11, pp. 7615–7631, 2020.

[80] A. P. Radünz, T. L. da Silveira, F. M. Bayer, and R. J. Cintra, "Data-independent low-complexity klt approximations for image and video coding," *Signal Processing: Image Communication*, vol. 101, p. 116 585, 2022.

[81] N. Kouadria, K. Mechouek, D. Messadeg, and N. Doghmane, "Pruned discrete tchebichef transform for image coding in wireless multimedia sensor networks," *AEU-International Journal of Electronics and Communications*, vol. 74, pp. 123–127, 2017.

[82] V. A. Coutinho, R. J. Cintra, F. M. Bayer, P. A. Oliveira, R. S. Oliveira, and A. Madanayake, "Pruned discrete tchebichef transform approximation for image compression," *Circuits, Systems, and Signal Processing*, vol. 37, no. 10, pp. 4363–4383, 2018.

[83] G. Paim, L. M. G. Rocha, G. M. Santana, L. B. Soares, E. A. C. da Costa, and S. Bampi, "Power-, area-, and compression-efficient eight-point approximate 2-d discrete tchebichef transform hardware design combining truncation pruning and efficient transposition buffers," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 2, pp. 680–693, 2018.

[84] S. Farsiani and A. M. Sodagar, "Hardware and power-efficient compression technique based on discrete tchebichef transform for neural recording microsystems," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, IEEE, 2020, pp. 3489–3492.

[85] S. Farsiani and A. M. Sodagar, "Compact agile tchebycheff transform variant for temporal compression of neural signals on brain-implantable microsystems," *Integration*, vol. 90, pp. 171–182, 2023.

[86] S. Prattipati, S. Ishwar, M. Swamy, and P. K. Meher, "A fast 8× 8 integer tchebichef transform and comparison with integer cosine transform for image compression," in *2013 IEEE 56th international midwest symposium on circuits and systems (MWSCAS)*, IEEE, 2013, pp. 1294–1297.

[87] F. Ernawan, N. A. Abu, and N. Suryana, "Tmt quantization table generation based on psychovisual threshold for image compression," in *2013 International Conference of Information and Communication Technology (ICoICT)*.

[88] R. Mukundan, S. Ong, and P. A. Lee, "Image analysis by tchebichef moments," *IEEE Transactions on image Processing*, vol. 10, no. 9, pp. 1357–1364, 2001.

[89] B. Chen, G. Coatrieux, J. Wu, Z. Dong, J. L. Coatrieux, and H. Shu, "Fast computation of sliding discrete tchebichef moments and its application in duplicated regions detection," *IEEE Transactions on Signal Processing*, vol. 63, no. 20, pp. 5424–5436, 2015.

[90] F. Ernawan and M. N. Kabir, "An improved watermarking technique for copyright protection based on tchebichef moments," *IEEE Access*, vol. 7, pp. 151 985–152 003, 2019.

[91] H. Zhang, X. Dai, P. Sun, H. Zhu, and H. Shu, "Symmetric image recognition by tchebichef moment invariants," in *2010 IEEE International Conference on Image Processing*, IEEE, 2010, pp. 2273–2276.

[92] X. Ding, N. Zhu, L. Li, Y. Li, and G. Yang, "Robust localization of interpolated frames by motion-compensated frame interpolation based on an artifact indicated map and tchebichef moments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 7, pp. 1893–1906, 2018.

[93] M. Kiruba and V. Sumathy, "Register pre-allocation based folded discrete tchebichef transformation technique for image compression," *Integration*, vol. 60, pp. 13–24, 2018.

[94] P. A. Oliveira, R. J. Cintra, F. M. Bayer, S. Kulasekera, and A. Madanayake, "A discrete tchebichef transform approximation for image and video coding," *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1137–1141, 2015.

[95] A. Mefoued, S. Harize, and N. Kouadria, "Efficient, low complexity 8-point discrete tchebichef transform approximation for signal processing applications," *Journal of the Franklin Institute*, vol. 360, no. 7, pp. 4807–4829, 2023.

[96] G. Paim, G. M. Santana, L. M. G. Rocha, L. B. Soares, E. A. C. da Costa, and S. Bampi, "Exploring approximations in 4-and 8-point dtt hardware architectures for low-power image compression," *Analog Integrated Circuits and Signal Processing*, vol. 97, no. 3, pp. 503–514, 2018.

[97] G. Paim, L. B. Soares, R. Ferreira, E. Costa, and S. Bampi, "Pruning and approximation of coefficients for power-efficient 2-d discrete tchebichef transform," in *2017 15th IEEE International New Circuits and Systems Conference (NEWCAS)*, IEEE, 2017, pp. 25–28.

[98] H. Bateman, *Higher transcendental functions [volumes i-iii]*. McGraw-Hill Book Company, 1953, vol. 1.

[99] I. S. Fathi, M. A. Ahmed, and M. Makhlouf, "An efficient computation of discrete orthogonal moments for bio-signals reconstruction," *EURASIP Journal on Advances in Signal Processing*, vol. 2022, no. 1, pp. 1–24, 2022.

[100] S. Bouguezel, M. O. Ahmad, and M. Swamy, "Binary discrete cosine and hartley transforms," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 4, pp. 989–1002, 2012.

[101] L. S. d. Oliveira and S. F. Saramago, "Multiobjective optimization techniques applied to engineering problems," *Journal of the brazilian society of mechanical sciences and engineering*, vol. 32, pp. 94–105, 2010.

[102] G. A. Seber, *A matrix handbook for statisticians*. John Wiley & Sons, 2008.

[103] A. Mefoued, N. Kouadria, S. Harize, and N. Doghmane, "Improved discrete tchebichef transform approximations for efficient image compression," *Journal of Real-Time Image Processing*, vol. 21, no. 1, p. 12, 2024.

[104] D. S. Watkins, *Fundamentals of matrix computations*. John Wiley & Sons, 2004.

[105] K. Karhunen, "Under lineare methoden in der wahr scheinlichkeitsrechnung," *Annales Academiae Scientiarun Fennicae Series A1: Mathematia Physica*, vol. 47, 1947.

[106] J. Mitchell, "Digital compression and coding of continuous-tone still images: Requirements and guidelines," *ITU-T Recommendation T*, vol. 81, 1992.

[107] 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, 2012.

[108] B. Silveira *et al.*, "Power-efficient sum of absolute differences hardware architecture using adder compressors for integer motion estimation design," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 12, pp. 3126–3137, 2017.

[109] C. Tablada, F. M. Bayer, and R. J. Cintra, "A class of dct approximations based on the feig–winograd algorithm," *Signal Processing*, vol. 113, pp. 38–51, 2015.

[110] R. Thukral, A. K. Aggarwal, A. S. Arora, T. Dora, and S. Sancheti, "Artificial intelligence-based prediction of oral mucositis in patients with head-and-neck cancer: A prospective observational study utilizing a thermographic approach," *Cancer Research, Statistics, and Treatment*, vol. 6, no. 2, pp. 181–190, 2023.

[111] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," pp. 770–778, 2016.

[112] M. Khodaee, T. Hwang, J. Kim, S. P. Norman, S. M. Robeson, and C. Song, "Monitoring forest infestation and fire disturbance in the southern appalachian using a time series analysis of landsat imagery," *Remote Sensing*, vol. 12, no. 15, p. 2412, 2020.

[113] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, and S. Homayouni, "Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 6308–6325, 2020.

[114] F. H. Shajin, P. Rajesh, and M. R. Raja, "An efficient vlsi architecture for fast motion estimation exploiting zero motion prejudgment technique and a new quadrant-based search algorithm in hevc," *Circuits, Systems, and Signal Processing*, pp. 1–24, 2022.

[115] B. Bross *et al.*, "Overview of the versatile video coding (vvc) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.

[116] A. Diana Andrushia and R. Thangarjan, "Saliency-based image compression using walsh–hadamard transform (wht)," *Biologically rationalized computing techniques for image processing applications*, pp. 21–42, 2018.

[117] R. Dony *et al.*, "Karhunen-loeve transform," *The transform and data compression handbook*, vol. 1, no. 1-34, p. 29, 2001.

[118] G. Lakhani, "Modified JPEG Huffman Coding," *IEEE transactions on image processing*, vol. 12, no. 2, pp. 159–169, 2003.

[119] E.-h. Yang and L. Wang, "Joint optimization of run-length coding, huffman coding, and quantization table with complete baseline jpeg decoder compatibility," *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 63–74, 2008.

[120] V. Ratnakar and M. Livny, "An efficient algorithm for optimizing dct quantization," *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 267–270, 2000.

[121] 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, 2001.

[122] E.-H. Yang, C. Sun, and J. Meng, "Quantization table design revisited for image/video coding," *IEEE Transactions on image processing*, vol. 23, no. 11, pp. 4799–4811, 2014.

[123] X. Yan, Y. Fan, K. Chen, X. Yu, and X. Zeng, "Qnet: An adaptive quantization table generator based on convolutional neural network," *IEEE Transactions on Image Processing*, vol. 29, pp. 9654–9664, 2020.

[124] M. Hopkins, M. Mitzenmacher, and S. Wagner-Carena, "Simulated annealing for jpeg quantization," *arXiv preprint arXiv:1709.00649*, 2017.

[125] B. Xiao, W. Shi, G. Lu, and W. Li, "An optimized quantization technique for image compression using discrete tchebichef transform," *Pattern Recognition and Image Analysis*, vol. 28, pp. 371–378, 2018.

[126] S. Prattipati, M. Swamy, and P. K. Meher, "A variable quantization technique for image compression using integer tchebichef transform," pp. 1–5, 2013.

[127] M. Parker, "Chapter 25 - image and video compression fundamentals," in *Digital Signal Processing 101 (Second Edition)*, M. Parker, Ed., Second Edition, Newnes, 2017, pp. 329–346, ISBN: 978-0-12-811453-7. DOI: https://doi.org/10.1016/B978-0-12-811453-7.00025-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128114537000251.

[128] E. Kodak, "Kodak lossless true color image suite (photocd pcd0992) ⟨https://r0k.us/graphics/kodak/⟩," no. 5,