UNIVERSITY OF MISKOLC



Faculty of Mechanical Engineering and Informatics

Efficient Visual Data Encoding Methods for Robust Spatial Perception Robotic Systems

PH.D. DISSERTATION SUMMARY

Prepared by:

Omar Mohammed Salih Alsabaawi
M.Sc. in Computer Technology Engineering

József Hatvany Doctoral School for Computer Science and Engineering

Research area: Computer Supported Production Systems
Research group: Automatization and Control Systems

Head of doctoral school: **Prof. Dr. László Kovács**
Supervisor: **Prof. Dr. József Vásárhelyi**

Miskolc 2026

Contents

1 Introduction **1**
1.1 Background and Motivation 1
1.2 Problem Statement 2
1.3 Research Objectives 2

2 Baseline DFT-Domain Compression Method for Spatial Perception **5**
2.1 Introduction 5
2.2 The Proposed Compression Algorithm 5
2.2.1 Discrete Fourier Transform and Quantization 5
2.2.2 Coefficient Reduction Algorithm 6
2.2.3 Encoding of Zero and Nonzero Values 7
2.3 The Decompression Algorithm 7
2.4 Experimental Results 8
2.5 Discussion and Comparative Analysis 9

3 Beyond JPEG Limitations: Improved Image Compression for 3D Reconstruction **11**
3.1 Introduction 11
3.2 The Proposed Compression Pipeline 11
3.2.1 Discrete Cosine Transform 11
3.2.2 Optimized Quantization 11
3.2.3 First-Order Difference Processing 12
3.3 The Proposed Decompression Pipeline 12
3.4 Experimental Results 13

4 End-to-End Multi-Level Encoding for Robust Monocular Visual ORB-SLAM **15**
4.1 Introduction 15
4.2 The Proposed End-to-End Frame Encoding Pipeline 15
4.2.1 Multi-Level Encoding Module on the Mobile Robot 15
4.2.2 Decoding Module on the Edge Server 17
4.3 Experimental Results and Discussion 17

5 Visual Data Encoding in Multi-Robot Visual SLAM Systems **21**
5.1 Introduction 21
5.2 Proposed Multi-Robot V-SLAM System Architecture 21
5.2.1 Proposed Communication Module 22

5.3 Results and Discussion 23

6 Efficient FPGA Implementation of a Visual Data Encoding/Decoding Pipeline 27
6.1 Introduction 27
6.2 Hardware-Software Co-Design Strategy 27
6.2.1 Encoding and Decoding Pipeline 27
6.3 Implementation results and Discussion 28

7 Summary and Future Research Directions 31
7.1 Dissertation Summary 31
7.2 Future Research Directions 31
7.3 Theses 32
7.3.1 Thesis I 32
7.3.2 Thesis II 32
7.3.3 Thesis III 32
7.3.4 Thesis IV 32
7.3.5 Thesis V 32

Author's publications 33

References 35

1 Introduction

1.1 Background and Motivation

Computer vision enables robotic and autonomous systems to perform recognition, localization, mapping, motion planning, and decision-making [1]. Although visual sensing and three-dimensional maps improve environmental understanding, they impose substantial computational, storage, communication, and energy demands, particularly on resource-constrained platforms [2]. Reliable 3D reconstruction also requires the preservation of geometry, depth, structure, and surface continuity [3]. The value of visual data for robotic perception therefore depends not only on its visual appearance but also on its ability to support feature extraction, correspondence estimation, pose estimation, depth inference, and reconstruction. Traditional compression standards are primarily remove perceptually redundant information to preserve visual quality for human observers [4, 5]. However, visually insignificant information may be essential for Convolutional Neural Networks (CNNs), segmentation algorithms, Simultaneous Localization and Mapping (SLAM) front ends, and 3D reconstruction, while visually acceptable images may not retain the structures required for reliable machine analysis [5]. Compression should therefore be evaluated according to its effect on the target perception task rather than image appearance alone. Previous studies demonstrate that the effects of compression are task-dependent. Appropriate compression can reduce data size while maintaining classification and object-detection performance, whereas severe degradation can reduce model robustness [1, 6, 7, 8, 9]. Task-aware methods improve efficiency by preserving information relevant to machine analysis. These include saliency-guided compression for object detection [10], feature-level compression [11], and methods developed for biomedical detection, classification, and segmentation [12, 13, 14, 15]. This requirement is particularly important for geometric perception because compression artifacts and quantization errors can degrade feature matching, depth estimation, point-cloud generation, and surface reconstruction [16]. Jointly optimizing encoding and reconstruction can maintain reliable 3D recovery beyond the range achieved by conventional task-independent coding [3]. Accordingly, this dissertation develops efficient visual data encoding methods that reduce the storage, communication, and computational requirements of resource-constrained robotic systems while preserving the structural, feature-level, and geometric information required for VSLAM and 3D reconstruction.

1.2 Problem Statement

Autonomous robots, Visual Simultaneous Localization and Mapping (VSLAM) systems, intelligent transportation, industrial inspection, collaborative robotics, and Internet of Things (IoT)-based environments continuously generate large visual data streams for perception, decision-making, mapping, and 3D reconstruction. Processing and transmitting these data create substantial storage, communication, computational, and latency challenges, particularly under limited bandwidth, hardware, and energy resources [17]. Traditional standards such as JPEG, JPEG2000, HEVC, and VVC primarily optimize compression for human visual perception. Consequently, visually acceptable reconstructions may not preserve the semantic and geometric information required by machine-vision tasks. Learned image compression has improved rate-distortion performance through nonlinear transforms, hyperprior entropy models [18], autoregressive context models with hierarchical priors [19], and attention-based contextual entropy estimation with Gaussian mixture models [20]. Although these developments have established Neural Image Compression (NIC) as an important alternative to conventional codecs, many compression methods remain human-oriented or require computationally intensive learned models. Edges, corners, textures, feature correspondences, and spatial gradients may appear visually insignificant but remain essential for localization, visual odometry, feature matching, segmentation, and 3D reconstruction [21]. Distortion of these structures during transform coding and quantization can therefore degrade robotic perception despite acceptable reconstructed-image quality. Furthermore, many recent image-, feature-, and scalable-coding approaches remain task-specific or unsuitable for resource-constrained robotic platforms. This dissertation therefore investigates transform-domain visual data encoding methods that provide deterministic execution, low computational complexity, FPGA-oriented implementation, and preservation of spatial perception performance. The proposed methods complement learned compression approaches while addressing the practical requirements of real-time, edge-assisted robotic systems.

1.3 Research Objectives

The primary objective of this dissertation is to develop efficient visual data encoding and decoding methods that reduce the computational, memory, storage, and communication requirements of resource-constrained robotic perception systems while preserving reliable localization, mapping, and 3D reconstruction. The specific objectives are to:

- Develop efficient Discrete Cosine Transform (DCT)- and Discrete Fourier Transform (DFT)-based compression methods that achieve high compression ratios while preserving structural and feature-level information.
- Apply the proposed methods to structured-light and multi-view 3D reconstruction and evaluate the trade-off between compression efficiency and reconstruction accuracy.
- Evaluate the effects of compression on trajectory estimation, mapping consistency, feature preservation, and hardware-resource utilization, in addition to conventional image-quality metrics.
- Integrate the visual data encoding methods into monocular Oriented FAST and Rotated BRIEF-Simultaneous Localization and Mapping (ORB-SLAM) through a multi-level framework based on transform-domain processing, quantization, coefficient reduction, and compact representation.
- Develop an edge-assisted multi-robot VSLAM architecture that partitions the computation between mobile robots and edge devices while supporting map fusion, loop closure detection, and global trajectory optimization under limited bandwidth.
- Implement and validate the encoding and decoding framework on a Field-Programmable Gate Array (FPGA)-based embedded platform to assess its feasibility for resource-constrained robotic systems.
- Compare the proposed methods with JPEG and JPEG2000 standards using image-quality metrics and robotic perception indicators.

2 Baseline DFT-Domain Compression Method for Spatial Perception

2.1 Introduction

The DFT is less commonly used in practical compression standards because its coefficients contain both real and imaginary components, increasing processing and storage requirements. Nevertheless, the DFT provides a complete frequency-domain representation and clearly separates low- and high-frequency information. It can also be efficiently implemented using the Fast Fourier Transform (FFT) and is well supported by platforms such as MATLAB Simulink and AMD Vivado/Vitis, making it suitable for FPGA-based hardware–software co-design. To address the limitations of DFT-based compression, this chapter introduces three original contributions:

- Integration of the 2D DFT into a matrix-reduction framework, extending previous DCT- and Discrete wavelet transform (DWT)-based approaches to the DFT domain.
- Replacement of the limited-data probability lookup table introduced in [22, 23] with minimum and maximum bounding parameters, thereby reducing header overhead and improving the compression ratio.
- Development of a novel zero-elimination technique for encoding high-frequency coefficient matrices.

2.2 The Proposed Compression Algorithm

The proposed method is illustrated in Figure 2.1. The input image is divided, beginning at its top-left corner, into non-overlapping $m \times n$ blocks. Each block is then processed through the DFT-based compression pipeline to generate the final compressed bitstream.

2.2.1 Discrete Fourier Transform and Quantization

In the proposed method, the DFT is applied independently to each 4×4 image block. The resulting real and imaginary coefficient matrices are uniformly quantized by dividing each coefficient by a heuristically selected Quantization factor (Q) and rounding the result to the nearest integer. Quantization increases the occurrence of small or zero-valued high-frequency coefficients, reducing

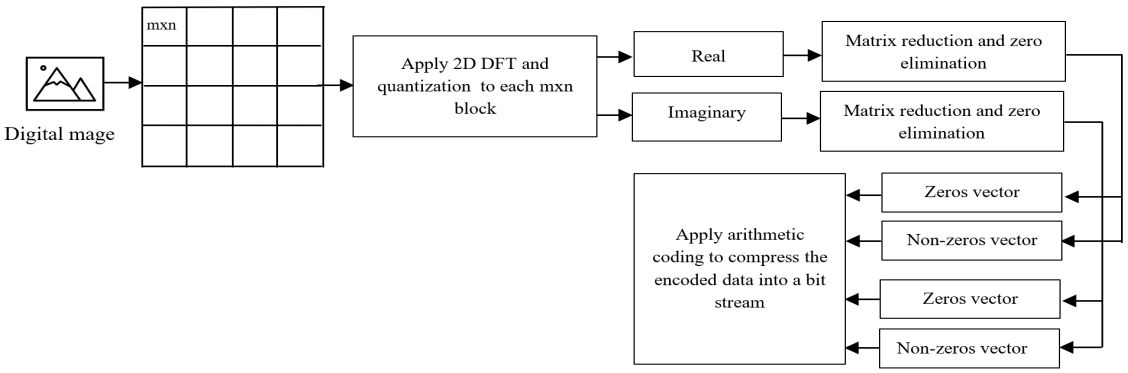


Figure 2.1: The proposed DFT-based compression method.

their representation requirements and improving the compression ratio. The DC coefficient of each real-component block is extracted and stored in a Low-Frequency Coefficient matrix (LFC matrix), while its original position is replaced by zero. The remaining real coefficients and all imaginary coefficients form the corresponding High-Frequency Coefficient matrices (HFC matrices).

2.2.2 Coefficient Reduction Algorithm

Because the HFC matrices are substantially larger than the LFC matrix, they are further compressed using a proposed coefficient-reduction algorithm. Every three consecutive coefficients are combined into one encoded value using predefined random weights:

$$E_i = \sum_{j=1}^{3} C_j K_j, \tag{2.1}$$

where E_i is the ith encoded value, C_j is the jth coefficient, and K_j is its corresponding random weight or encoding key. The index j ranges from 1 to 3, while i ranges from 1 to $MN/3$. If necessary, the coefficient sequence is zero-padded to the nearest multiple of three. Figure 2.2 illustrates this reduction process. During decompression, the coefficient triplet corresponding to each encoded value is recovered through a bounded search:

$$(\hat{C}_{i,1}, \hat{C}_{i,2}, \hat{C}_{i,3}) = \left\{ (C_1, C_2, C_3) \in [C_{\min}, C_{\max}] \;\middle|\; E_i = \sum_{j=1}^{3} C_j K_j \right\} \tag{2.2}$$

where $(\hat{C}_{i,1}, \hat{C}_{i,2}, \hat{C}_{i,3})$ is the reconstructed coefficient triplet associated with E_i, while $C_{\min}$ and $C_{\max}$ define the search bounds. Unlike the conventional approach, which stores a complete array of possible coefficient values, the proposed method stores only these two bounds in the compressed

header. This reduces header overhead and restricts the number of combinations examined during decoding.

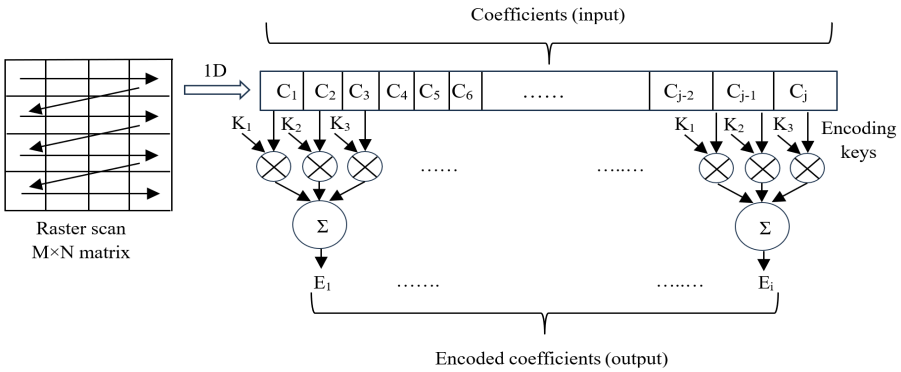


Figure 2.2: Coefficient reduction for an $M \times N$ matrix.

2.2.3 Encoding of Zero and Nonzero Values

Following coefficient reduction, the HFC matrices contain many zero values. To exploit this redundancy, zero and nonzero coefficients are separated before arithmetic coding. The nonzero coefficients are stored sequentially in a Value Matrix, while a Zero Matrix records the number of zeros occurring between successive nonzero coefficients. This representation preserves the original coefficient positions while avoiding the explicit storage of repeated zeros. The same procedure is applied to the LFC matrix. The resulting Value and Zero matrices are then compressed using arithmetic coding and transmitted with the information required for decompression.

2.3 The Decompression Algorithm

The decompression algorithm reverses the operations performed during compression. Arithmetic decoding first recovers the LFC, Value, and Zero matrices. The Value and Zero matrices are then combined by inserting each nonzero coefficient into its original position and restoring the recorded number of zeros between successive coefficients. This process regenerates the reduced HFC matrices for the real and imaginary components. A sequential search algorithm subsequently reconstructs the three original coefficients associated with each encoded value E_i. The algorithm examines candidate triplets within the range $[C_{\min}, C_{\max}]$ until it identifies a combination satisfying Equation 2.2. Here, K_1, K_2, and K_3 are the encoding keys used during coefficient reduction. Because the minimum and maximum coefficient values are retrieved from the compressed header, the search is confined to a bounded space. Figure 2.3 illustrates this process. After the HFC and LFC matrices have been restored, the real and imaginary DFT components are reconstructed.

Inverse quantization and the inverse DFT are then applied to each block to obtain the decompressed image. The complete decompression pipeline is presented in Figure 2.4.

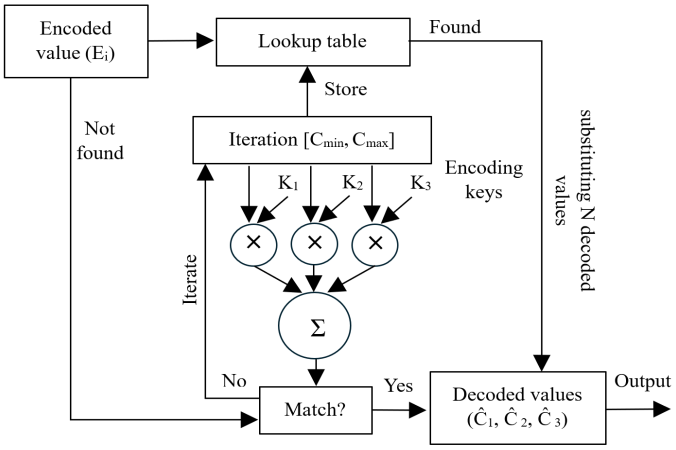


Figure 2.3: Sequential search mechanism used during decompression.

2.4 Experimental Results

The proposed compression and decompression method was evaluated on grayscale and color images of different sizes using RMSE and PSNR. For the grayscale images, the compressed sizes ranged from 47.8 to 228 KB, with PSNR values between 39.0 and 47.3 dB. For the color images, the compressed sizes ranged from 182.8 to 437.4 KB, while the PSNR values ranged from 40.6 to 44.9 dB. Increasing the quantization level reduced the compressed size and bit rate at the expense of higher RMSE and lower PSNR. The method was also evaluated using structured-light images for 3D surface reconstruction.

The employed scanner projects a stripe pattern onto the scene and captures it with a camera. The relationship between the camera and light source enables the patterned regions of a single 2D image to be converted into a 3D surface mesh within a few milliseconds. Structured-light images provide

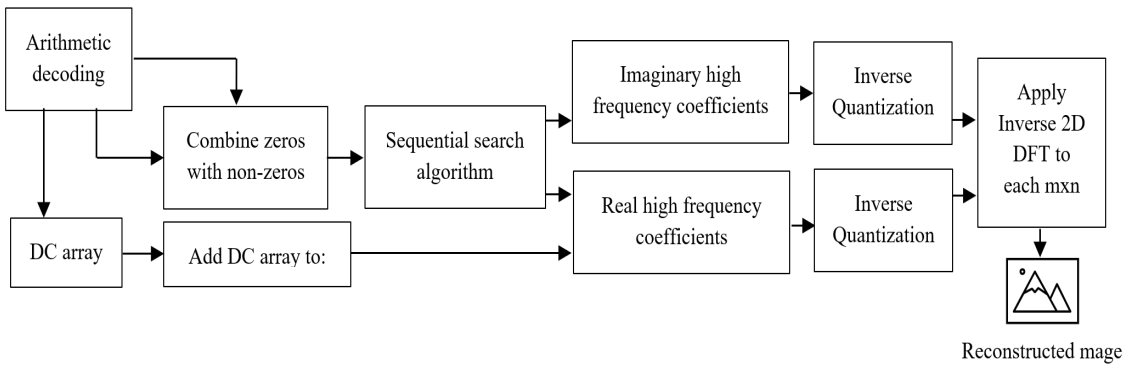


Figure 2.4: The proposed decompression process.

a demanding evaluation because compression artifacts affecting the projected stripe patterns can cause reconstruction failure or visible defects in the resulting surface. The structured light images were compressed and decompressed using the proposed method before being processed by the GMPR reconstruction algorithm.

2.5 Discussion and Comparative Analysis

The proposed method produces smaller compressed sizes and lower bit-per-pixel values than the DFT alone. Although aggressive high-frequency reduction increases RMSE and decreases PSNR in some cases, the reconstructed images maintain acceptable visual quality while achieving compression ratios of up to 98%. The proposed method was also compared with JPEG, which employs the DCT and Huffman coding, whereas the proposed approach combines the DFT, matrix reduction, and zero/non-zero coding. For structured-light images, the proposed method achieves better compression performance and preserves the stripe information required for 3D mesh reconstruction. In contrast, JPEG compression introduces degradation that can produce defective reconstructed surfaces. The proposed approach was further compared with previous matrix-reduction methods based on DWT and DCT. These methods perform compression and encryption simultaneously but require relatively complex compression and decompression procedures. The proposed method reduces the number of decompression steps and executes faster than the conventional matrix-reduction approach on the same hardware. It also requires less header information, thereby improving the compression ratio. These contributions enable faster reconstruction from highly compressed data while maintaining competitive RMSE and PSNR values for both conventional and structured-light images.

3 Beyond JPEG Limitations: Improved Image Compression for 3D Reconstruction

3.1 Introduction

This chapter presents a novel lossy image-compression algorithm based on a single-level DCT pipeline that integrates an optimized quantization matrix, DC differencing, zero-elimination, and arithmetic coding. Furthermore, the proposed method extends existing DCT-based approaches by addressing their limitations in compressing multi-view and structured-light images used for 3D surface reconstruction.

3.2 The Proposed Compression Pipeline

The proposed compression pipeline is illustrated in Figure 3.1. The input image is divided into non-overlapping 8×8 blocks to enable direct comparison with JPEG. If the image dimensions are not multiples of eight, zero-padding is applied. Each block is then processed according to the stages described below.

3.2.1 Discrete Cosine Transform

The DCT is applied to each 8×8 block, producing 64 frequency-domain coefficients. The first coefficient is the low-frequency DC component, whereas the remaining 63 are AC coefficients representing higher-frequency image information.

3.2.2 Optimized Quantization

The proposed quantization matrix was derived from the standard JPEG luminance quantization matrix and modified to preserve structurally important low-frequency information. The entries corresponding to the DC coefficient and its neighboring low-frequency AC coefficients were set to one, preventing these coefficients from being altered during quantization. Preserving them helps retain the image structures required for edges, corners, and feature extraction while the stronger quantization of higher-frequency components supports high compression ratios. The complete proposed pipeline is shown in Figure 3.1.

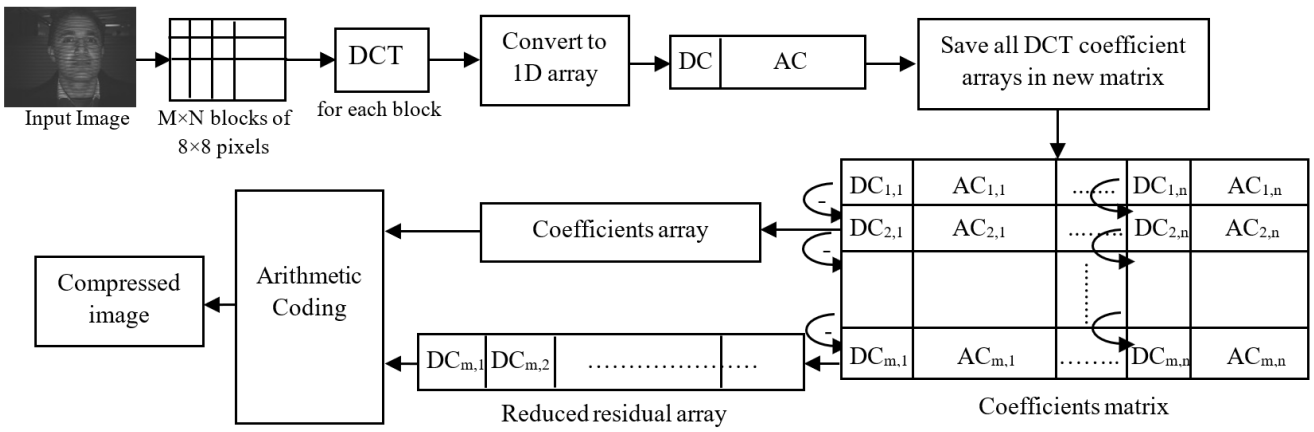


Figure 3.1: The proposed DCT-based compression pipeline.

3.2.3 First-Order Difference Processing

To reduce the magnitude of the DC coefficients, first-order differencing is applied independently to each column of image blocks. Each DC coefficient is replaced by the difference between its current value and the value in the following block row:

$$\Delta DC_{i,j} = DC_{i,j} - DC_{i+1,j} \qquad i = 1 : m-1, \quad j = 1 : n, \tag{3.1}$$

where m and n denote the numbers of block rows and columns, respectively. Because neighboring blocks generally have correlated DC values, the resulting differences typically have smaller magnitudes and can be encoded more efficiently. The final DC coefficient in each column is retained as a reference value and stored in a reduced residual array, while its position in the coefficient matrix is set to zero. The quantized coefficient matrix is then scanned block by block. A block containing only zeros is represented by a single zero in the coefficient array. Otherwise, it is represented by a header value of one followed by its 64 coefficients, as illustrated in Figure 3.2. Finally, the coefficient array and reduced residual array are compressed using arithmetic coding to produce the compressed image file.

3.3 The Proposed Decompression Pipeline

The decompression pipeline reverses the compression operations, as illustrated in Figure 3.3. Arithmetic decoding first recovers the coefficient array and reduced residual array. The quantized coefficient matrix is then reconstructed by scanning the coefficient array. A header value of one indicates that the following 64 values form a nonzero 8×8 block, whereas a zero indicates an all-zero block.

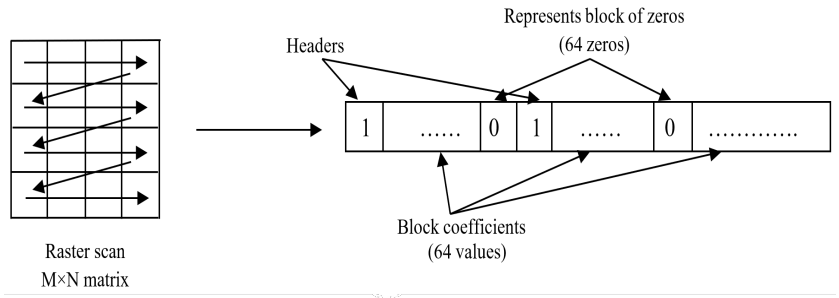


Figure 3.2: Format of the coefficient array.

At this stage, the stored DC values represent the differences between consecutive block rows. The final DC coefficient in each column is first restored from the reduced residual array. The remaining coefficients are then reconstructed in reverse row order using

$$DC_{i,j} = \Delta DC_{i,j} + DC_{i+1,j} \qquad i = m-1:1, \quad j = 1:n. \tag{3.2}$$

After restoring the DC coefficients, inverse quantization and the inverse DCT are applied to each block. The reconstructed blocks are subsequently combined, and any padding introduced during compression is removed to obtain the decompressed image.

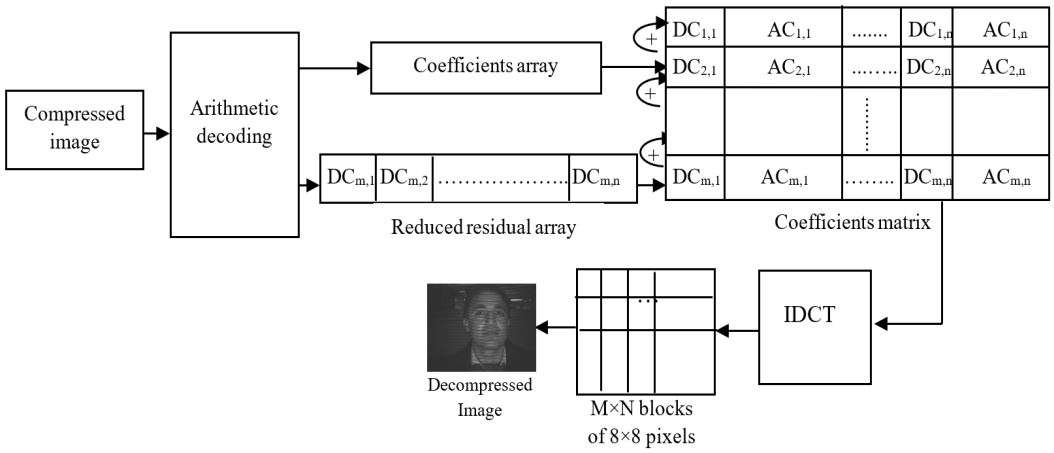


Figure 3.3: The proposed DCT-based decompression pipeline.

3.4 Experimental Results

The proposed algorithm was evaluated using three grayscale structured-light images having 1.38 MB size for each. The images were compressed to between 14 and 21.1 KB, with RMSE values of 3.3-8.3 and PSNR values of 38.9-42.9 dB. The algorithm was also applied to two multi-view image sequences used to generate 3D objects. The objects were reconstructed using 3DF Zephyr and

converted into meshes using Autodesk MeshMixer. Two multi-viewpoint image sets are used; the first is a 48-frame sequence which was reduced from 114.8 to 3.1 MB, while the second is a 42-frame sequence and was reduced from 100.38 to 2.39 MB. The proposed method was compared with JPEG using structured-light and multi-view images. At compression ratios of 98%-99%, JPEG produced substantial artifacts and missing surface regions. The proposed method preserved the image details required for 3D reconstruction at these higher compression ratios. Regarding multi-viewpoint images, at high compression ratios JPEG introduced artifacts and missing surface information into both reconstructed meshes. The proposed method achieved compression ratios of up to 99%, whereas JPEG at 97% produced corrupted meshes and failed to support reliable reconstruction.

4 End-to-End Multi-Level Encoding for Robust Monocular Visual ORB-SLAM

4.1 Introduction

This chapter presents a novel multi-level visual data encoding method integrated into monocular Oriented FAST and Rotated BRIEF Simultaneous Localization and Mapping (ORB-SLAM). The method extends the proposed matrix-reduction framework to achieve high compression ratios while preserving the visual information required for reliable robotic perception. Its integration is evaluated within both Compress-Then-Analyze (CTA) and Analyze-Then-Compress (ATC) VSLAM architectures. Furthermore, the evaluation investigates the effects of lossy compression on compressed data size, image reconstruction quality, execution time, and trajectory-estimation accuracy.

4.2 The Proposed End-to-End Frame Encoding Pipeline

4.2.1 Multi-Level Encoding Module on the Mobile Robot

The encoding module is deployed on the mobile robot and can operate within both CTA and ATC architectures. In CTA, every camera frame is compressed and transmitted to the edge server, where it is decoded and processed by the complete ORB-SLAM pipeline. In ATC, tracking is performed locally using the raw frames, while encoded keyframes are transmitted to the edge server for map reconstruction, loop-closure detection, and global optimization. The appropriate architecture depends on the available onboard computational resources and application requirements. The proposed encoding pipeline is illustrated in Figure 4.1. Each input frame is divided into 8×8 blocks and transformed into the frequency domain using the DCT. Each block produces one DC coefficient, representing its average intensity, and 63 AC coefficients representing its frequency details. The coefficients are then quantized using the optimized quantization matrix introduced in Section 3, which preserves low-frequency structural information important for ORB feature extraction and matching. Following quantization, the coefficients are arranged into a one-dimensional sequence using zigzag scanning. The proposed multi-level encoding algorithm then applies successive coefficient-reduction stages according to a user-defined number of levels. At each level, three coefficients are combined into one encoded value using:

$$E_i^{(\ell)} = \sum_{j=1}^{3} C_{i,j}^{(\ell)} K_j^{(\ell)}, \tag{4.1}$$

where $E_i^{(\ell)}$ is the ith encoded value at level ℓ, $C_{i,j}^{(\ell)}$ denotes the corresponding input coefficients, and $K_j^{(\ell)}$ represents the encoding keys. In this implementation, the same keys K_1, K_2, and K_3 are reused at every level to simplify the system, although different key sets could also be employed. Figure 4.2 illustrates the multi-level encoding architecture employing a two-level configuration.

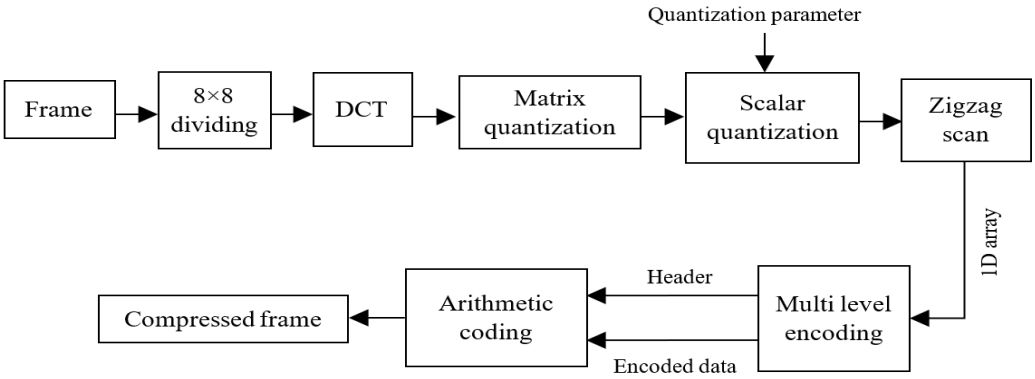


Figure 4.1: The proposed multi-level image-encoding pipeline.

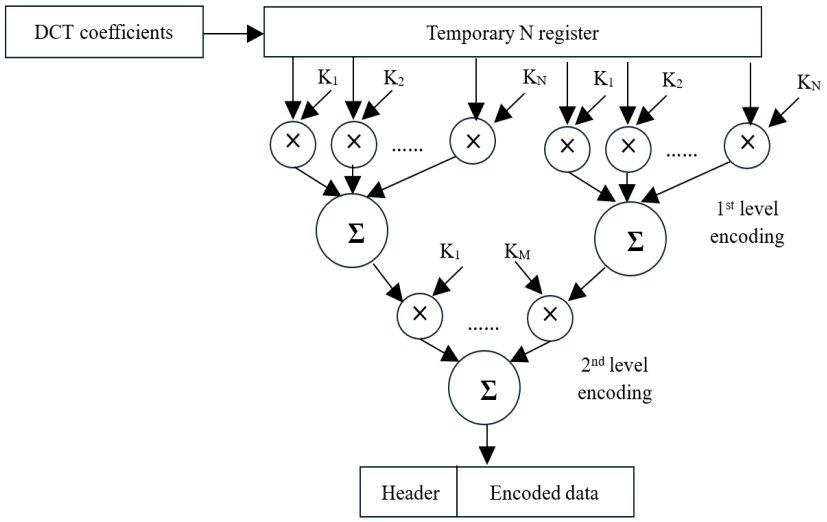


Figure 4.2: Two-level encoding with the same key set K_i reused at both levels.

The method extends the matrix-reduction algorithm presented Section 2.2.2 by applying it successively over multiple levels. The coefficient-reduction stages are lossless with respect to their quantized inputs; therefore, additional encoding levels reduce the representation size without introducing distortion beyond that caused by quantization. Duplicate encoded values are removed, and the resulting data are subsequently compressed using arithmetic coding. In CTA, the compressed frames are transmitted to the edge server for the complete ORB-SLAM process. In ATC,

only compressed keyframes are transmitted for mapping, loop-closure detection, and global optimization.

4.2.2 Decoding Module on the Edge Server

The decoding module is deployed on the edge server and reverses the encoding operations, as illustrated in Figure 4.3. Arithmetic decoding first recovers the encoded coefficient data and any retained header information. Sequential-search decoding is adopted to prioritize compression efficiency while exploiting the greater computational capability of the edge server. After reconstructing the coefficients at each encoding level in reverse order, inverse zigzag scanning, inverse quantization, and the inverse DCT are applied to recover the image frames required by the corresponding ORB-SLAM operations.

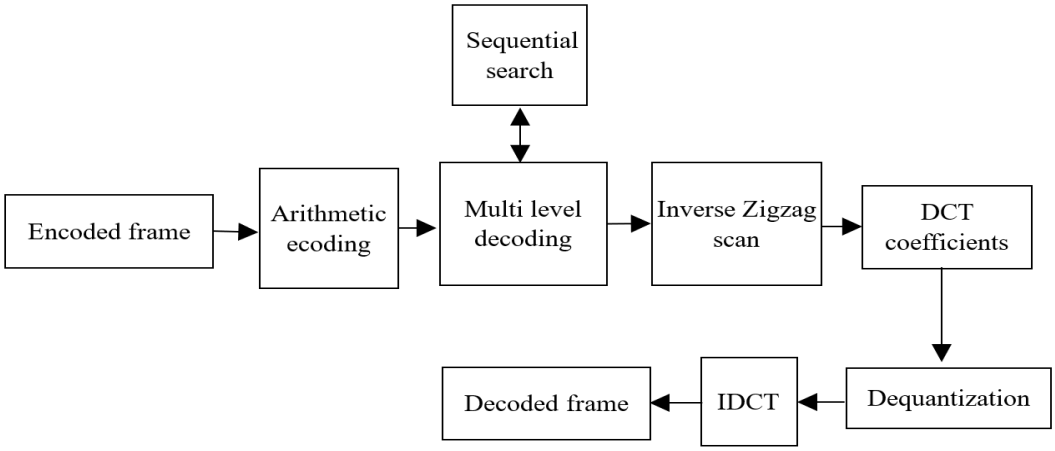


Figure 4.3: The proposed image-decoding pipeline.

4.3 Experimental Results and Discussion

Evaluation was conducted using the TUM dataset [24], which provides RGB images, depth maps, and ground-truth camera trajectories for VSLAM benchmarks. The "freiburg3 long office household" sequence contains 2585 frames captured at 30 fps with a resolution of 640×480 pixels. Its 87.09 s trajectory spans 5.12 m $\times$ 4.89 m $\times$ 0.54 m and includes an overlap between its beginning and end that enables loop-closure evaluation. The standard ORB-SLAM was first evaluated using uncompressed visual data, and the resulting trajectory RMSE was 0.22114. JPEG was then integrated as an end-to-end encoding and decoding framework in which the cloud module decodes the received frames and performs tracking, mapping, and loop closure. Quantization Factors (QFs) from 5 to 20 were applied to evaluate the trade-off between compression and trajectory estimation accuracy. The ORB-SLAM remained robust up to QF=15 with successful loop

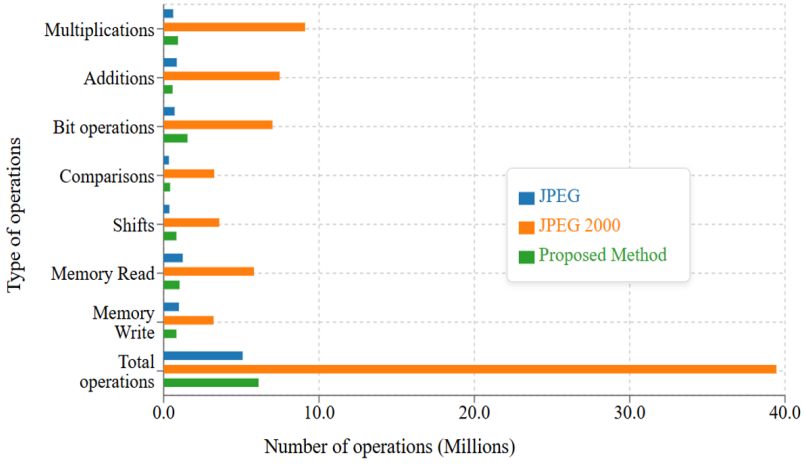
closure. At QF=20, compression artifacts degraded ORB feature detection and matching, causing trajectory drift and preventing completion of the map. The same cloud-based architecture was evaluated using JPEG2000, which controls compression through a target Compression Ratio (CR). The robust trajectory estimation was maintained up to CR=97%. At CR=98%, the system failed to detect loop closure. Finally, the proposed multi-level algorithm was integrated into ORB-SLAM as an end-to-end visual data encoding and decoding framework. Table 4.1 compares JPEG, JPEG2000, and the proposed method at the closest attainable compression ratios because JPEG and the proposed method use quantization parameters, whereas JPEG2000 directly specifies the target CR. JPEG2000 achieved the highest PSNR and SSIM, while the proposed method produced the smallest compressed file size with competitive reconstruction quality. Its longer processing time resulted from coefficient reduction and sequential-search operations. These times are implementation-dependent because the proposed method was validated in MATLAB, whereas JPEG2000 was evaluated using the optimized native-C OpenJPEG library. Software optimization or FPGA- and GPU-based acceleration can reduce this overhead, while the smaller transmitted data volume can reduce communication latency in edge-assisted VSLAM systems.

Table 4.1: Comparison of compression performance metrics between JPEG, JPEG2000, and the proposed methods at approximately equal compression ratios.

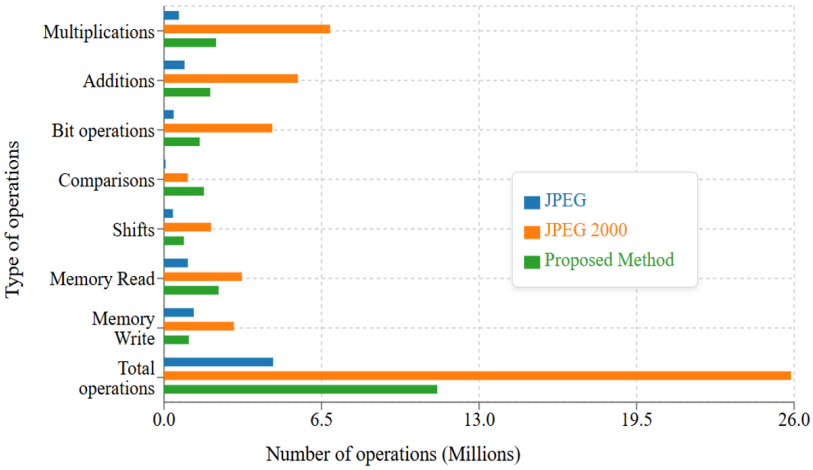
Method	**CR** [%]	**Size** [KB]	**PSNR** [dB]	**SSIM**	**Processing time** [s]	
					Encoding	**Decoding**
JPEG	85.7	81	26.1274	0.801	0.1921	0.3793
JPEG2000	85	72	40.09	0.9647	0.112	0.444
Proposed (single-level)	86	68	33.5493	0.9398	0.6951	0.6093
Proposed (two-level)		67			0.7252	0.6966

Unlike the JPEG-based system, the proposed method maintained robust pose estimation and successful loop closure for QF values from 5 to 20. It remained robust under more severe quantization up to QF=75 but failed to detect loop closure at QF=100. Increasing QF generated more repeated values and zeros, thereby reducing the encoded data and header sizes, increasing the compression ratio, and accelerating processing. Compared with JPEG2000, the proposed and JPEG2000-based systems exhibited similar trajectory-estimation behavior. The proposed method supported successful trajectory estimation at CR=96%, while JPEG2000 remained successful up to CR=97%, with slightly better reconstructed-frame quality and execution time. However, the

computational-complexity analysis in Figure 4.4 showed that the proposed encoder and decoder required fewer basic operations than JPEG2000 while providing comparable trajectory-estimation performance. JPEG was slightly less complex but produced substantially poorer performance at high compression levels. The proposed method therefore provides a practical compromise between compression efficiency, computational complexity, and ORB-SLAM robustness for mobile robotic systems. Across the evaluated methods, ORB-SLAM became less robust when PSNR fell below 30 dB and SSIM fell below 0.87.



a) Computational complexity of the encoding systems



b) Computational complexity comparison of the decoding systems

Figure 4.4: Computational complexity of the three implemented systems.

5 Visual Data Encoding in Multi-Robot Visual SLAM Systems

5.1 Introduction

Multi-robot systems combine measurements from multiple agents to extend sensing coverage, enable parallel task execution, and improve robustness through redundancy. However, collaborative VSLAM is constrained by the limited computational, storage, and communication resources of mobile robots. Although edge-assisted computing can reduce onboard processing requirements, transmitting large volumes of visual data over bandwidth-limited networks may degrade system performance. To address these challenges, this chapter presents a centralized edge-assisted ORB-SLAM framework comprising two mobile robots and an edge device. Each robot independently performs real-time tracking and local mapping, while the edge device handles global map fusion, loop-closure detection, and trajectory optimization. The main contributions of this chapter include the development of a centralized edge-assisted multi-robot architecture that extends the single-robot ORB-SLAM framework by distributing the principal VSLAM tasks between mobile robots and an edge device. The architecture incorporates an efficient visual-data encoding and communication module that reduces transmission requirements while preserving VSLAM accuracy. A comprehensive evaluation further examines the effects of lossy visual-data compression on transmission efficiency, trajectory-estimation accuracy, system robustness, and global-map consistency.

5.2 Proposed Multi-Robot V-SLAM System Architecture

A centralized edge-assisted multi-robot VSLAM system with global map reconstruction is proposed based on the Analyze-Then-Compress paradigm. The ORB-SLAM architecture is distributed between the mobile robots and the edge device. Each robot performs tracking and local mapping and uses a communication module to transmit data to the edge. The edge device performs map optimization, loop closure, and global map fusion using the local maps generated by the robots. Figure 5.1 illustrates the implemented architecture, which employs two robots as a case study but can be extended to support additional robots.

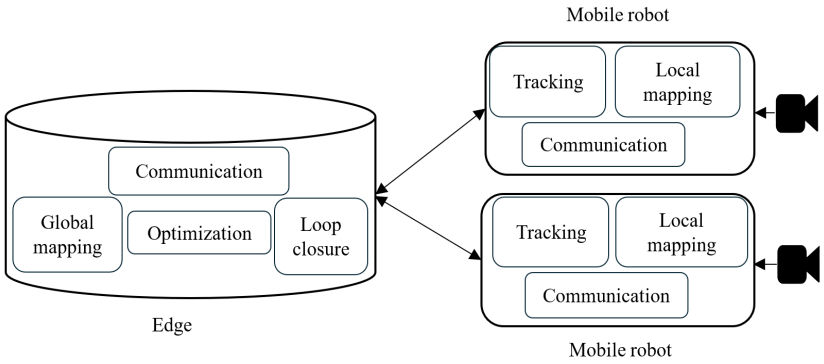


Figure 5.1: Proposed multi-robot ORB-SLAM architecture.

5.2.1 Proposed Communication Module

The proposed communication module comprises an encoder deployed on each mobile robot and a decoder implemented on the edge device. The encoder processes candidate keyframes generated by the tracking module and is designed for low complexity, low latency, and minimal resource consumption to support real-time operation on low-power mobile robots. The single-level encoding method introduced in Section 4, is further developed to improve compression efficiency and accelerate the reconstruction of the reduced DCT coefficients. Three decoding scenarios are investigated according to the inclusion or omission of the dictionary and sequential-search stages. In the sequential-search method, the dictionary attached to the compressed-frame header and the encoding keys are used to estimate the original DCT coefficients. A cache table stores the decoded values obtained for successful matches, allowing repeated encoded values to be reconstructed directly without additional searches. When no cached match is available, the search continues using other candidate values from the unique table. The direct-calculation method reverses the encoding process using division and modulo operations, without requiring a lookup table. Each encoded value represents three original coefficients combined through a weighted sum. To provide unique mappings for all possible coefficient triplets, the encoding keys are selected as powers of a base number greater than the maximum value in the coefficient data. The table method employs a large predefined table on the edge device. Each row associates an encoded value with its three corresponding original coefficients. During decoding, the received value is matched against the first column of the table, and the associated coefficient triplet is added to the reconstructed output.

5.3 Results and Discussion

Evaluation used the TUM RGB-D *freiburg3 long office household* dataset described in Section 4.3. Its 2585 frames were partitioned between two robots with sufficient overlap for global map fusion and loop-closure detection. Trajectory accuracy was evaluated using RMSE against the ground truth. The architecture was first evaluated using uncompressed frames. Each robot initialized its local map, tracked candidate keyframes, and transmitted them to the edge server, which optimized and fused the local maps. The results confirm reliable trajectory estimation and successful global map fusion. The communication module was then evaluated at quantization values from $Q = 20$ to $Q = 100$. The proposed system maintained reliable localization and map fusion throughout this range. At $Q = 100$, it reduced communication memory by 39%, 30%, and 34% for Robot-1, Robot-2, and the edge server, respectively, while producing RMSE values of 0.1472, 0.1276, and 0.1357 m. The corresponding global trajectory in Figure 5.2 remained consistent. At $Q = 120$, however, severe compression artifacts reduced ORB-feature matching accuracy, increased accumulated pose errors, and prevented reliable global map reconstruction.

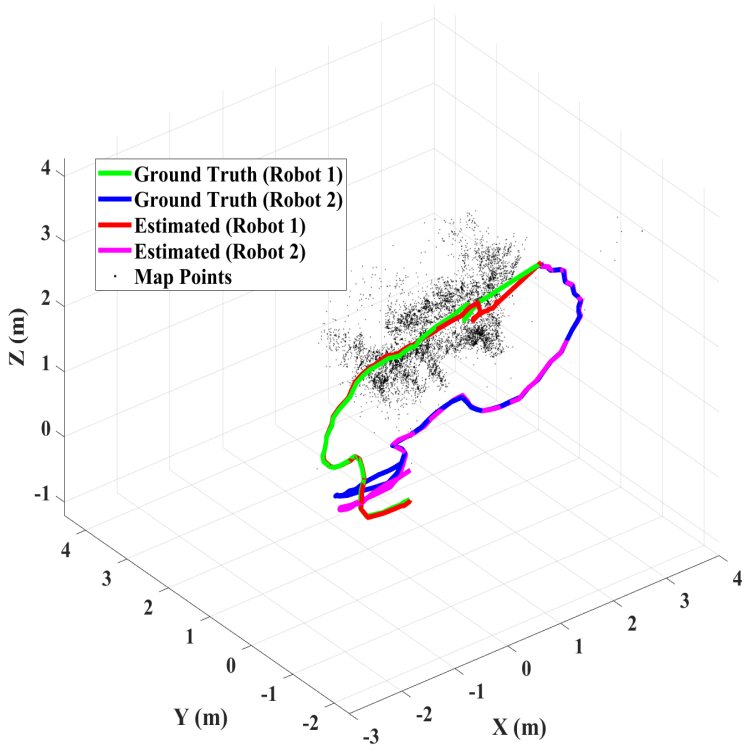


Figure 5.2: Visualization of global map trajectories at high quantization (Q=100).

The proposed architecture was compared with equivalent JPEG and JPEG2000-based systems. The JPEG system maintained stable operation only up to $Q = 60$ and subsequently failed to fuse the global map because compression artifacts degraded feature matching and pose estima-

tion. Although JPEG achieved greater memory reduction, Robot-2 reached an RMSE of 0.2741 m at $Q = 60$ and produced fewer map points than the proposed method, indicating a greater loss of geometric information. JPEG2000 preserved visual structures more effectively than JPEG, producing more map points and competitive localization accuracy. However, its communication-memory requirements frequently approached or exceeded the uncompressed reference size, limiting its suitability for bandwidth-constrained edge-assisted systems. The proposed method therefore provided a more balanced trade-off between communication efficiency, localization accuracy, and preservation of the geometric information required for global map fusion. Table 5.1 compares a representative 900 KB keyframe processed using JPEG, JPEG2000, and the three coefficient-reconstruction implementations of the proposed method: sequential searching, direct calculation decoding, and table lookup decoding. The searching method includes unique probability information in the compressed bitstream, whereas direct calculation and table lookup reconstruct the coefficients using the encoding keys without this additional header information. Consequently, the latter methods produce smaller compressed sizes and higher compression ratios.

The proposed method achieved higher compression ratios and better reconstructed-frame quality than JPEG, although JPEG required less encoding time because it does not perform DCT-coefficient minimization. JPEG2000 provided better reconstructed-frame quality at comparable compression ratios. The proposed searching method achieved slightly lower compression ratios than JPEG2000, whereas direct calculation and table lookup reached up to 99.1% by eliminating the unique-table header. Direct calculation decoded the data faster than table lookup because it used integer arithmetic without accessing a large, memory-intensive mapping table. The table implementation required repeated indexing operations, resulting in reduced cache efficiency and a constant decoding time of 18.2558 s. This result is implementation-dependent and may differ in optimized software or hardware realizations. The complexity analysis showed that JPEG required the fewest basic operations but performed poorly at high compression ratios. The proposed searching and direct-calculation implementations required fewer computational resources than JPEG2000, while JPEG2000 and the proposed table-based implementation were more resource-intensive. Overall, the searching and direct-calculation variants provide a practical compromise between compression efficiency, computational complexity, communication requirements, and ORB-SLAM robustness for resource-constrained mobile robots.

Table 5.1: Compression performance comparison using frame 2068 (original size = 900 KB).

Q	**Compressed size (KB)**	**PSNR**	**SSIM**	**CR (%)**	**Execution Time (s)**	
					Encoding	**Decoding**
JPEG						
5	169	33.7052	0.9271	81.2	0.3647	0.5487
10	165	32.2014	0.9150	81.6	0.3165	0.5325
15	161	31.2274	0.8948	82.1	0.2964	0.5414
20	152	28.6571	0.8346	83.1	0.2571	0.5021
JPEG2000						
n/a	189	41.23	0.9815	85	0.165	0.455
n/a	153	38.45	0.9406	88	0.124	0.432
n/a	45	34.71	0.9125	95	0.131	0.411
n/a	18	31.12	0.8849	98	0.148	0.427
Proposed methods: searching, direct calculation, and table (reported from top to bottom for each Q value)						
25	162	30.3663	0.9032	82	0.4425	0.8704
	86.4			90.4	0.5881	0.8731
						18.2558
50	135	29.4914	0.8738	85	0.4221	0.7321
	54.9			93.9	0.4843	0.6469
						18.2558
75	81	28.6856	0.8451	91	0.3827	0.6053
	23.4			97.4	0.4346	0.5375
						18.2558
100	21.6	27.8886	0.8157	97.6	0.3670	0.5357
	8.1			99.1	0.4038	0.5120
						18.2558

6 Efficient FPGA Implementation of a Visual Data Encoding/Decoding Pipeline

6.1 Introduction

This chapter presents a complete FPGA-based hardware–software co-design that partitions processing between the programmable logic and processing system. Computationally intensive operations are hardware-accelerated, while control and entropy coding remain software-based to preserve flexibility. Integrated into an edge-assisted indoor navigation framework, the system enables efficient visual data reduction and supports real-time processing.

6.2 Hardware-Software Co-Design Strategy

Processing units are partitioned between the PL and PS according to computational regularity, data parallelism, and control-flow complexity. Deterministic, fixed-size, and highly parallel operations are accelerated in the PL using HLS optimizations, whereas irregular or serial tasks involving complex control, dynamic data structures, dictionary-based hash tables, or range-coding state machines are executed in the PS.

6.2.1 Encoding and Decoding Pipeline

The proposed end-to-end system, illustrated in Figure 6.1, compresses RGB candidate keyframes for transmission within the analyze-then-compress VSLAM framework. Computationally intensive operations are implemented as Vitis HLS-generated AXI4 IP cores in the PL, while entropy coding and protocol framing are executed on the Arm Cortex-A53 PS. This hardware-software co-design reduces wireless bandwidth requirements while preserving trajectory accuracy and image fidelity, evaluated using RMSE, PSNR, and SSIM.

Input keyframes are divided into non-overlapping 8×8 blocks and processed using a fixed-point 2D DCT implemented as consecutive row-wise and column-wise 1D transforms. The resulting coefficients are quantized using a runtime-programmable matrix configured through AXI4-Lite, allowing compression quality to be adjusted without FPGA re-synthesis. Zigzag scanning orders the coefficients from low to high frequency, while DPCM encodes differences between adjacent

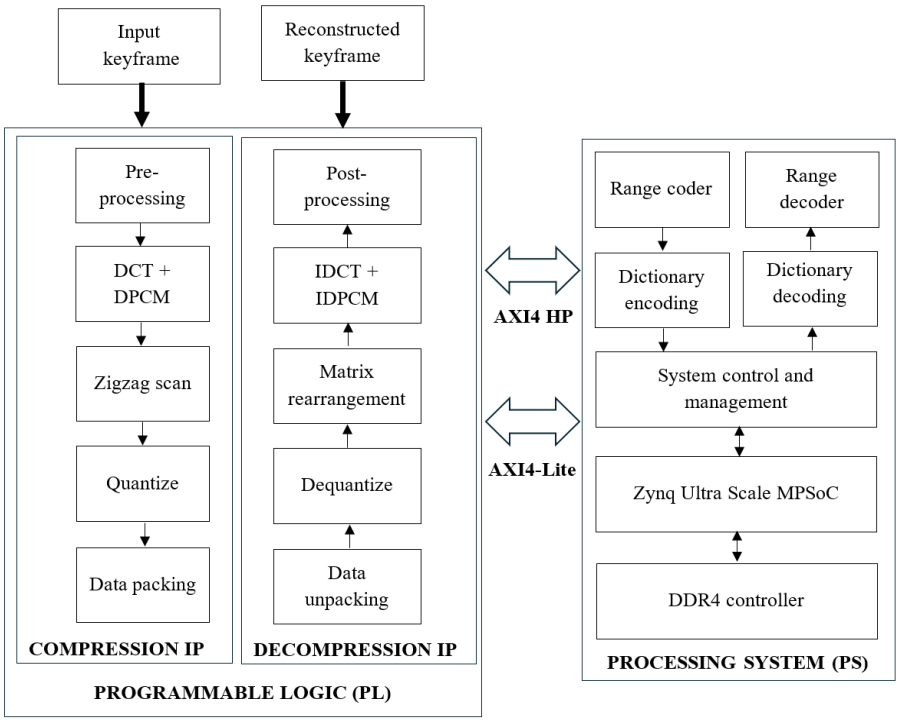


Figure 6.1: Hardware–software co-design architecture of the proposed FPGA-based visual data encoding and decoding system.

DC coefficients. A 63-bit mask identifies the non-zero AC positions within a 64-bit word that includes the DC position, and only non-zero coefficients are retained. Every three coefficients are then packed into a fixed-field word; thus, a block containing 12 non-zero coefficients requires four packed words instead of 64 values. The packed stream is transferred to the PS, where a frequency-ordered Dictionary Table (DICT) represents common words using compact indices and handles infrequent symbols through an escape (ESC) mechanism. A binary-scaled integer range coder produces the final compressed bitstream. Decompression reverses these operations through range decoding and inverse dictionary mapping. Each IP core communicates with the PS through AXI4-Lite for control and configuration and an AXI4 full master interface for 256-bit burst DMA transfers. The implemented FPGA architecture is shown in Figure 6.2.

6.3 Implementation results and Discussion

The proposed system was evaluated on the AMD Kria KV260 Vision AI Starter Kit (XCK26 UltraScale+ MPSoC) using the TUM RGB-D benchmark and compared with CPU-based JPEG and JPEG2000 implementations. The dataset contains 2585 timestamped 640×480 RGB images captured at 30 fps. Its 87.09 s trajectory covers 5.12m×4.89m×0.54m and includes a loop, enabling evaluation of loop closure and trajectory accuracy against motion-capture ground truth.

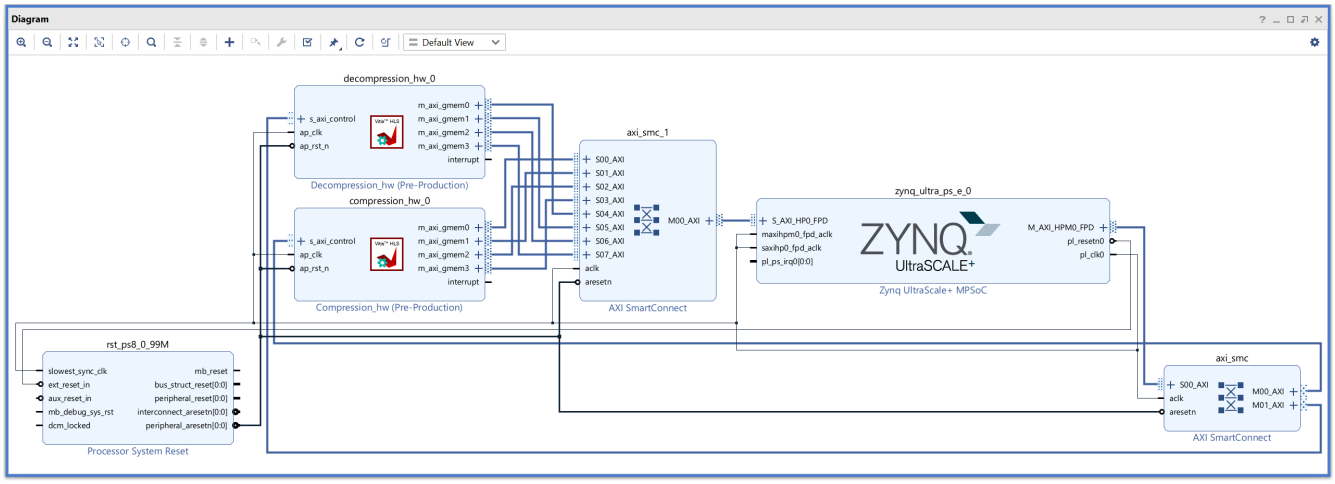


Figure 6.2: Implementation of the proposed FPGA-based system using AMD Vivado.

The fixed-point encoder and decoder were implemented using FPGA-based hardware–software co-design, demonstrating a symmetric FPGA-to-FPGA architecture between the mobile robot and edge server. JPEG and JPEG2000 were implemented in MATLAB on an Intel Core i7-12700H CPU with 32 GB RAM. Performance was evaluated using the percentage compression ratio, PSNR, SSIM, encoding and decoding times, and ORB-SLAM trajectory RMSE. JPEG provides moderate compression but exhibits substantial image-quality degradation as CR increases. JPEG2000 achieves higher compression while preserving superior PSNR and SSIM. The proposed FPGA method produces lower image-fidelity metrics than JPEG2000 because of its fixed-point implementation, but maintains acceptable quality and progressively shorter processing times as compression increases. It therefore provides a practical trade-off between image quality, data reduction, and computational efficiency. Figure 6.3 shows that the RMSE of the trajectory increases with CR for all methods. JPEG is restricted to approximately 68-71% CR and is comparatively sensitive to compression. JPEG2000 maintains a lower RMSE in a wider range, whereas the proposed method generally achieves a lower RMSE than JPEG2000 at approximately 79-89% CR. Above 90%, however, its RMSE increases more rapidly and approaches or slightly exceeds that of JPEG2000, indicating reduced navigation accuracy under extreme compression. In addition to compression performance and navigation accuracy, the hardware efficiency of the proposed implementation was evaluated in terms of resource utilization and power consumption, as presented in Figures 6.4. Both units exhibit low resource utilization. The encoder uses 12% of the DSP slices and 11% of the LUTs, while the decoder uses 6% of the DSP slices, 10% of the LUTs, and 7% of the flip-flops. This leaves sufficient hardware capacity for further system integration. Total power consumption is approximately 2.9 W for both units, with dynamic and static power accounting for approximately

90% and 10%, respectively. The PS contributes more than 90% of the dynamic power, whereas the PL consumes only a small proportion. The decoder has slightly lower dynamic power because of its lower computational demand.

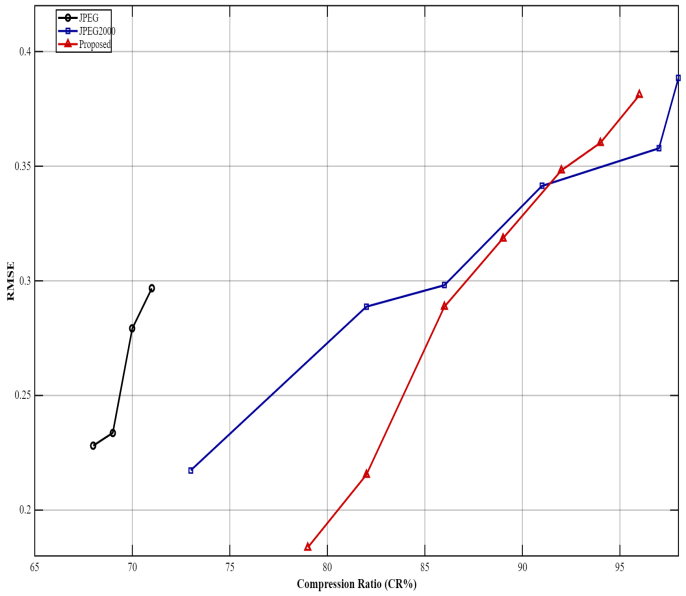
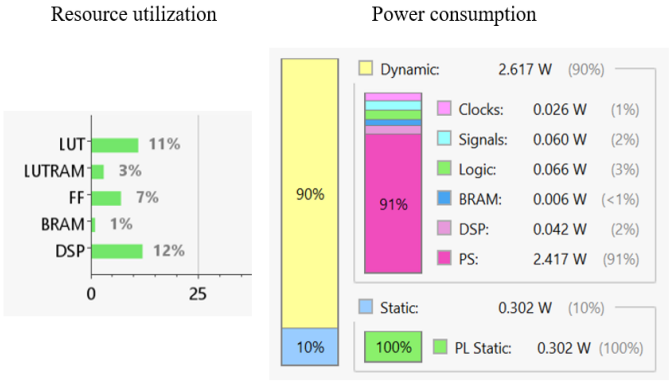
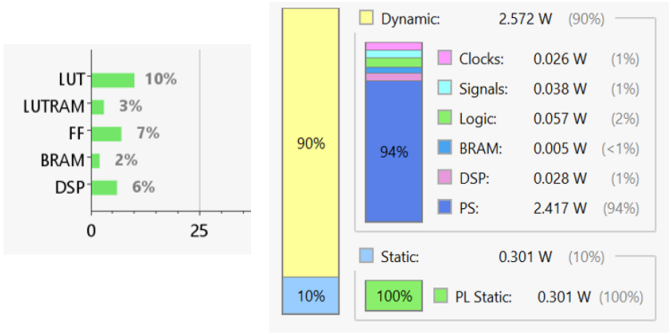


Figure 6.3: Graph of RMSE for VSLAM trajectory vs. compression ratio of the three implemented encoding systems.



(a) Encoder circuit (robot side)



(b) Decoder circuit (edge side)

Figure 6.4: FPGA-based system implementation: (a) Resource utilization and (b) power consumption.

7 Summary and Future Research Directions

7.1 Dissertation Summary

This dissertation develops efficient visual data encoding and compression methods for VSLAM and 3D reconstruction in bandwidth- and resource-constrained environments. A multi-level encoding scheme is integrated into monocular ORB-SLAM to balance compression efficiency and feature preservation while maintaining localization and mapping accuracy under different quantization levels. Additionally, an edge-assisted multi-robot VSLAM architecture partitions computational tasks between mobile robots and a centralized edge device. Its encoding and communication module reduces memory and communication requirements while supporting collaborative map fusion and consistent global localization. For 3D reconstruction, novel DFT- and DCT-based methods combine optimized quantization, coefficient reduction, matrix reduction, zero elimination, and entropy coding to compress structured-light and multi-view images while preserving the geometric information required for accurate mesh reconstruction. Finally, an FPGA-based hardware-software co-design implements the proposed compression pipeline on an embedded platform. The results demonstrate efficient resource utilization and the feasibility of the proposed methods for scalable, resource-constrained robotic perception systems.

7.2 Future Research Directions

Future research should evaluate the proposed methods under more diverse environmental, motion, and communication conditions and develop adaptive architectures that adjust compression, keyframe selection, and task partitioning according to network, energy, motion, and mapping requirements. The multi-robot VSLAM architecture could be extended to larger heterogeneous teams through scalable map fusion, distributed loop closure, and communication-aware coordination. Further applications may include industrial and agricultural inspection, search-and-rescue operations, cultural heritage digitization, medical imaging, and robotic manipulation. Controlled comparisons with representative learned compression models under identical experimental conditions are also required to assess their rate-distortion performance against the deterministic execution, computational efficiency, and FPGA suitability of the proposed transform-domain methods.

7.3 Theses

7.3.1 Thesis I. (Related Publications: [P15], [P11], [P12])

I introduced a novel image compression algorithm based on the two-dimensional Discrete Fourier Transform (2D DFT), incorporating enhanced coefficient reduction and zero elimination to achieve high compression ratios while preserving reconstructed-image quality.

7.3.2 Thesis II. (Related Publications: [P13], [P14], [P8])

I proposed a novel lossy image compression algorithm optimized for 3D mesh reconstruction from structured-light and multi-view images, addressing the limitations of JPEG at high compression ratios. The method enhances DCT-based compression through an optimized quantization matrix, DC differencing, zero elimination, and arithmetic coding.

7.3.3 Thesis III. (Related Publications: [P3], [P6], [P7])

I developed and implemented a novel multi-level visual data encoding method for robust monocular ORB-SLAM in cloud–edge computing environments. The method reduces bandwidth requirements while preserving feature-detection quality and trajectory-estimation accuracy.

7.3.4 Thesis IV. (Related Publications: [P4], [P5], [P2])

I introduced a centralized edge-assisted architecture for multi-robot Visual SLAM that addresses the computational and resource limitations of fully onboard deployment. The framework assigns real-time tracking and local mapping to the robots while offloading global optimization, map fusion, and loop closure detection to an edge device.

7.3.5 Thesis V. (Related Publications: [P1], [P9], [P10])

I proposed and implemented an FPGA-based hardware-software co-design framework for visual data compression, with optimized task partitioning between the Programmable Logic (PL) and Processing System (PS).

Author's publications

[P1] O. M. Salih and J. Vásárhelyi, 'A hardware–software fpga co-design of a visual data compression pipeline for indoor edge-assisted navigation,' in *2026 27th International Carpathian Control Conference (ICCC)*, IEEE, 2026, pp. 1–6.

Scopus indexed.

[P2] H. Rostum, O. M. Salih and J. Vásárhelyi, 'Efficient real-time drone mapping with visual slam and python,' *Pollack Periodica*, pp. 606–2025. 2026,

Scopus [Q3].

[P3] O. M. Salih and J. Vásárhelyi, 'End-to-end multi-level encoding methods of visual data compression for robust monocular visual orb-slam,' *Acta Polytechnica Hungarica*, vol. 22, no. 5, 2025,

Scopus [Q2], WoS (SCIE) [Q2].

[P4] O. M. Salih and J. Vásárhelyi, 'Efficient architecture for multi-robot visual slam systems,' in *2025 26th International Carpathian Control Conference (ICCC)*, IEEE, 2025, pp. 1–6.

Scopus indexed.

[P5] V. József and S. M. Omar, 'Peremhálózat alapú vslam keretrendszer erőforrás-korlátozott platformokhoz: An edge assisted vslam framework for resource constraint platforms,' *Energetika-Elektrotechnika–Számítástechnika és Oktatás Multi-konferencia*, pp. 178–182, 2025.

[P6] O. M. Salih, H. Rostum and J. Vásárhelyi, 'A novel method to improve the efficiency and performance of cloud-based visual simultaneous localization and mapping,' *Engineering Proceedings*, vol. 79, no. 1, p. 78, 2024,

Scopus [Q3].

[P7] O. M. Salih and J. Vásárhelyi, 'Visual data compression approaches for edge-based orb-vslam systems,' in *2024 25th International Carpathian Control Conference (ICCC)*, IEEE, 2024, pp. 1–6.

Scopus indexed.

[P8] O. M. Salih and J. Vásárhelyi, 'Visual data compression for machine vision applications,' in *Doktoranduszok fóruma: Gépészmérnöki és Informatikai Kar szekciókiadványa Conference : Miskolc Egyetemváros, Hungary 2023.11.23. - 2023,11,23*, University of Miskolc, 2024, pp. 128–133.

[P9] J. Vásárhelyi, O. M. Salih, H. M. Rostum and R. Benotsname, 'An overview of energies problems in robotic systems,' *Energies*, vol. 16, no. 24, p. 8060, 2023,

Scopus [Q1], WoS (SCIE) [Q1].

[P10] M. R. Hussam, M. S. Omar and H. Noha, 'Fpga implementation in mobile robot applications: State of the art review,' *Multidiszciplináris Tudományok*, vol. 13, no. 2, pp. 232–249, 2023.

[P11] O. M. Salih, H. M. Rostum, J. Vásárhelyi and M. M. Siddeq, 'Fast joint image compression-encryption algorithm used for 3d reconstruction,' in *2023 24th International Carpathian Control Conference (ICCC)*, IEEE, 2023, pp. 400–405.

Scopus indexed.

[P12] O. M. Salih, 'Quick sequential search algorithm used to decode high frequency matrices,' *International Journal of Electrical and Computer Engineering*, 2023.

[P13] O. M. Salih, M. H. Rasheed, M. M. Siddeq and M. A. Rodrigues, 'Image compression for quality 3d reconstruction,' *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 5, pp. 2271–2287, 2022,

Scopus [Q1], WoS (SCIE) [Q1].

[P14] M. H. Rasheed, O. M. Salih and M. M. Siddeq, 'Joint image encryption and compression schemes based on hexa-coding,' *Periodicals of Engineering and Natural Sciences*, vol. 9, no. 2, pp. 569–580. 2021,

Scopus [Q2].

[P15] M. H. Rasheed, O. M. Salih, M. M. Siddeq and M. A. Rodrigues, 'Image compression based on 2d discrete fourier transform and matrix minimization algorithm,' *Array*, vol. 6, p. 100 024, 2020,

WoS (ESCI) [Q1].

References

[1] C. R. Steffens, L. R. V. Messias, P. L. J. Drews and S. S. da Costa Botelho, 'Can exposure, noise and compression affect image recognition? an assessment of the impacts on state-of-the-art convnets,' in *2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE)*, IEEE, 2019, pp. 61–66.

[2] A. Junaedy, H. Masuta, K. Sawai, T. Motoyoshi and N. Takagi, 'Real-time 3d map building in a mobile robot system with low-bandwidth communication,' *Robotics*, vol. 12, no. 6, p. 157, 2023.

[3] G. Alex and Y. Schechner Yoav, 'Image compression optimized for 3d reconstruction by utilizing deep neural networks [j],' *Journal of Visual Communication and Image Representation*, vol. 79, 2021.

[4] Z. Liu et al., 'Deepn-jpeg: A deep neural network favorable jpeg-based image compression framework,' in *Proceedings of the 55th annual design automation conference*, 2018, pp. 1–6.

[5] Z. Liu et al., 'Machine vision guided 3d medical image compression for efficient transmission and accurate segmentation in the clouds,' in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 687–12 696.

[6] M. Dejean-Servières, K. Desnos, K. Abdelouahab, W. Hamidouche, L. Morin and M. Pelcat, 'Study of the impact of standard image compression techniques on performance of image classification with a convolutional neural network,' Ph.D. dissertation, INSA Rennes; Univ Rennes; IETR; Institut Pascal, 2017.

[7] E.-H. Yang, H. Amer and Y. Jiang, 'Compression helps deep learning in image classification,' *Entropy*, vol. 23, no. 7, p. 881, 2021.

[8] K. Grm, V. Štruc, A. Artiges, M. Caron and H. K. Ekenel, 'Strengths and weaknesses of deep learning models for face recognition against image degradations,' *Iet Biometrics*, vol. 7, no. 1, pp. 81–89, 2018.

[9] I. Bouderbal, A. Amamra and M. A. Benatia, 'How would image down-sampling and compression impact object detection in the context of self-driving vehicles?' In *International Conference on Computing Systems and Applications*, Springer, 2020, pp. 25–37.

[10] H. Xiong and Y. Xu, 'Saliency-guided learned image compression for object detection,' in *International Conference on Neural Information Processing*, Springer, 2022, pp. 324–335.

[11] J. Zhong, T. Lei, G. Yao and P. Jiang, 'Robust aircraft detection with a simple and efficient model,' *Information*, vol. 9, no. 4, p. 74, 2018.

[12] E. S. Paulo, L. A. Thomaz, L. M. Távora, P. A. Assuncao and S. M. Faria, 'Extending the compression range of biomedical images for machine vision analysis,' in *2022 30th European Signal Processing Conference (EUSIPCO)*, IEEE, 2022, pp. 1273–1277.

[13] Y.-Y. Jo et al., 'Impact of image compression on deep learning-based mammogram classification,' *Scientific Reports*, vol. 11, no. 1, p. 7924, 2021.

[14] F. Ghazvinian Zanjani, S. Zinger, B. Piepers, S. Mahmoudpour, P. Schelkens and P. H. de With, 'Impact of jpeg 2000 compression on deep convolutional neural networks for metastatic cancer detection in histopathological images,' *Journal of Medical Imaging*, vol. 6, no. 2, pp. 027 501–027 501, 2019.

[15] B. Cardone and F. Di Martino, 'Bit reduced fcm with block fuzzy transforms for massive image segmentation,' *Information*, vol. 11, no. 7, p. 351, 2020.

[16] Y. M. Mostafa, M. N. Al-Berry and H. A. Shedeed, '3d reconstruction from jpg images.,' *International Journal of Advanced Computer Science & Applications*, vol. 16, no. 7, 2025.

[17] W. Yang, H. Huang, Y. Hu, L.-Y. Duan and J. Liu, 'Video coding for machines: Compact visual representation compression for intelligent collaborative analytics,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 7, pp. 5174–5191, 2024.

[18] J. Ballé, D. Minnen, S. Singh, S. J. Hwang and N. Johnston, 'Variational image compression with a scale hyperprior,' *arXiv preprint arXiv:1802.01436*, 2018.

[19] D. Minnen, J. Ballé and G. D. Toderici, 'Joint autoregressive and hierarchical priors for learned image compression,' *Advances in neural information processing systems*, vol. 31, 2018.

[20] Z. Cheng, H. Sun, M. Takeuchi and J. Katto, 'Learned image compression with discretized gaussian mixture likelihoods and attention modules,' in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7939–7948.

[21] J. Cao, X. Yao, H. Zhang, J. Jin, Y. Zhang and B. W.-K. Ling, 'Slimmable multi-task image compression for human and machine vision,' *IEEE Access*, vol. 11, pp. 29 946–29 958, 2023.

[22] M. Siddeq and M. Rodrigues, 'A new 2d image compression technique for 3d surface reconstruction,' in *Proceedings of 18th International Conference on Computers (part of CSCC'14). Recent advances in computer engineering series, 1 (22)*, World Scientific, Engineering Academy and Society (WSEAS), 2014, pp. 379–386.

[23] M. Siddeq and M. Rodrigues, 'A novel image compression algorithm for high resolution 3d reconstruction,' *3D Research*, vol. 5, no. 2, p. 7, 2014.

[24] J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers, 'A benchmark for the evaluation of rgb-d slam systems,' in *2012 IEEE/RSJ international conference on intelligent robots and systems*, IEEE, 2012, pp. 573–580.