DOI: 10.52756/ijeim.2025.v01.i01.001 Int. J. Eng. Infor. Manag. 1(1): 1-14 (2025)

International Journal of Engineering and Information Management
Journal homepage: www.ijeim.in

Efficient Image Synchronization and Change Detection Using XOR-Based PATCH Algorithms in Digital Twin Technology

Chandan Mukherjee[1], **Shiladitya Munshi**[2*], **Debasmita Das**[2], **Debankana Debnath**[2], **Diganta Das**[2]

[1]Department of Digital Technologies, American University of Phnom Penh, Cambodia
[2]Department of Computer Science & Engineering, Techno India University, Tripura, Agartala, India

***Corresponding Author:** shiladitya.explorer@gmail.com

Article Information

Type of Article: Original
Received: Nov 15, 2024
Accepted: Jan 5, 2025
Published: Jan 15, 2025

Keywords:
Change Detection of Image
Digital Twin
Image Processing
XOR operations.

Cite this article:
Chandan Mukherjee, Shiladitya Munshi, Debasmita Das, Debankana Debnath, and Diganta Das. (2025). Efficient Image Synchronization and Change Detection Using XOR-Based PATCH Algorithms in Digital Twin Technology. *International Journal of Engineering and Information Management*, 1(1), 1-14.

Abstract

This paper proposes an efficient XOR-based PATCH algorithm for real-time image synchronization and change detection in Digital Twin Technology. The primary objective is to minimize computational overhead, memory usage, and bandwidth by transmitting only pixel-level changes (PATCH) between physical and digital twins rather than entire image datasets. The proposed algorithm demonstrates versatility by effectively handling black and white, greyscale, and coloured images. Extensions of the algorithm for greyscale and coloured images involve applying XOR operations across intensity levels and RGB channels, respectively, ensuring minimal storage requirements while maintaining synchronization accuracy. Experimental results show that the algorithm scales linearly with image size, significantly improving bandwidth and memory efficiency, particularly in high resolution images with sparse changes. The scalability and adaptability of the XOR-based PATCH algorithm make it highly suitable for real-time applications in resource constrained environments, such as IoT-based remote monitoring systems.

1 Introduction

The rising adoption of Digital Twin Technology in manufacturing, urban planning, and healthcare has necessitated advancements in real-time image processing techniques. A digital twin, a virtual representation of a physical entity, requires continuous synchronization with its real-world counterpart to maintain accuracy and effectiveness. In this process, image data plays a crucial role in reflecting the real-time state of the physical asset. For example, in manufacturing, a digital twin must frequently update its image data to capture any physical equipment or infrastructure changes, ranging from minor adjustments to significant alterations.

Similarly, digital twins of organs or patients' medical conditions in healthcare require continuous monitoring to reflect up-to-date diagnostics and treatment responses. However, transmitting and processing high-resolution images in real time create significant challenges, particularly in bandwidth-constrained environments, such as Internet of Things (IoT) networks, which are commonly used for remote monitoring and control. The sheer size of these image files and the need for frequent updates lead to inefficiencies in data transmission and storage, limiting the scalability of digital twin systems.

The rapid advancement of digital twin technology has significantly transformed various industries, enabling seamless integration of physical and virtual environments (Wang et al., 2023). One particularly critical aspect of digital twin development is the ability to efficiently synchronize and detect changes between the physical and virtual representations.

This research paper explores the use of XOR-based PATCH algorithms to address the challenges of image synchronization and change detection in the context of digital twin technology. The integration of AI-powered technologies, such as computer vision (Pramanik et al., 2021), machine learning (Sarkar et al., 2019), and deep learning (Sarkar et al., 2022), has been a driving force behind the evolution of digital twin capabilities (Bankins et al., 2023). The dynamic nature of digital twins requires robust mechanisms for maintaining the synchronization between the physical and virtual counterparts, ensuring that any changes or updates in the physical world are accurately reflected in the digital twin.

The introduction of XOR-based PATCH algorithms, a novel approach to image processing, has the potential to revolutionize the way digital twins handle image synchronization and change detection. Digital twins have emerged as a transformative technology, enabling enhanced efficiency, predictability, and optimization across various industries, from manufacturing to healthcare (Homaei et al., 2023; Fuller et al., 2020).

The ability to create a virtual replica of a physical system, process, or asset allows for advanced simulations, predictive maintenance, and real-time monitoring, ultimately improving decision-making and operational effectiveness (Thelen et al., 2022). The XOR-based PATCH algorithm, a unique approach to image processing, offers several advantages in the context of digital twin technology (Diskin et al., 2014). The algorithm's ability to efficiently detect and synchronize changes between the physical and virtual environments is a crucial aspect of maintaining the integrity and accuracy of digital twins.

Furthermore, the integration of AI-powered tools, such as computer vision and deep learning, enhances the capabilities of digital twins, enabling more accurate diagnostics, predictive analytics, and personalized interventions. Efficient change detection and image synchronization are critical for maintaining accurate digital twins, especially as the size and resolution of image data increase. Traditional methods entail transmitting full images every time an update occurs (Trauer et al., 2020).

While these approaches are both bandwidth intensive and delay sensitive, real-time synchronization is a challenge (Lamboray et al., 2004). The above methods are particularly discouraged when the channel capacity is low or when frequent transmission of massive data is to be made. To address these challenges, this paper proposes a new XOR-based PATCH algorithm, which only sends the actual changes (delta) between the physical asset and its twins, much smaller than sending the full image. This is made possible by pixel level differences which results in optimum bandwidth use, reduction in processing time and optimal memory utilization. This makes the method suitable for use in several applications, including those that may be sensitive to resource utilization.

The main objectives of this study are as follows:

- To design an XOR-based PATCH generation algorithm that effectively identifies pixel-level differences between pictures of a physical asset and its virtual counterpart. It has been pointed out that for any practical applications in which the developed algorithm to be used in real-time systems, the algorithm must be relatively light and consume relatively limited computational resources.

- To develop an XOR-based PATCH application algorithm to update the digital twin with the changes (PATCH) to match the physical system with low computational and memory complexity. The proposed algorithm must be able to work with black and white, greyscale and coloured images to prove its flexibility in many areas of applications.

- To compare the proposed algorithms with the traditional ones regarding the essential criteria, including memory consumption, response time and synchronization accuracy, and bandwidth utilization. Further, a comparison of the proposed XOR-based image synchronization technique with the existing methods shall be performed to showcase the improvements.

- To learn more about additional possibilities and future uses for this technique, possible areas for its implementation, such as manufacturing, health care, or smart city, were examined. Thus, the scalability of the proposed method for different industries will be tested to show its generality.

The increasing complexity of digital twin systems across different sectors necessitates robust algorithms that handle a diverse range of image data. For instance, in innovative city management, high-resolution images from surveillance cameras, environmental monitoring sensors, and urban infrastructure simulations must be synchronized in real-time to ensure effective decision-making.

In healthcare, real-time updates to patient data through medical imaging require rapid processing to support accurate diagnostics and timely interventions. These applications emphasize the need for efficient, scalable, and lightweight algorithms like the proposed XOR-based PATCH, which can operate within the resource constraints of real-world systems while maintaining high accuracy in detecting changes.

Driven by the aforementioned objectives, the prime contributions of the current study may be summarized as follows:

- A novel application of XOR logic in digital twin image processing for efficient change detection and synchronization. This approach provides a lightweight alternative to deep learning models, which, while powerful, can be computationally expensive and unsuitable for environments with limited resources.

- An in-depth evaluation of the proposed algorithm's performance on real-time image data, covering different image types (black and white, greyscale, and coloured) and varying resolutions to demonstrate its effectiveness in diverse scenarios.

- Exploration of future extensions to 3D image processing and more complex digital twin environments. While the current work focuses on 2D image synchronization, the methods developed here lay the foundation for future research in 3D digital twins. These are becoming increasingly relevant in fields such as virtual reality, engineering, and architecture.

The rest of the article is structured as follows: Section 2 reviews existing techniques for image processing and change detection in digital twins, positioning the current study within the broader context of image synchronization methods. This review includes conventional approaches such as deep learning models, difference-based techniques, and heuristic-based methods, and it evaluates their respective advantages and limitations.

Next, Section 3 presents the XOR-based PATCH generation and application algorithms, detailing their design, implementation, and optimizations for memory and computational efficiency. Special attention is given to the algorithm's adaptability to different image types (black & white, greyscale, and coloured), showcasing its versatility in real-world applications.

Section 4 outlines the experimental methodology, assessment parameters and success criteria, thus affirming the success of the presented methodology compared to conventional approaches. It also discusses how the experimental insights would generalize to practical Digital Twin systems, identifying limitations inherent to the algorithm, including the ability to handle other transformations beyond affine, and the extension of the algorithm to 3D models, among others.

Lastly, Section 5 summarizes the findings and contributions of this work to discuss possible implications of this work in advancing the state of knowledge in Digital Twin Technology and its applicability in real-world sectors like manufacturing, healthcare, and smart cities that may benefit from real-time synchronization.

2 Literature Review

Digital Twin technology has been a topic of discussion in the last few years, being widely used in different industries including manufacturing, urban construction, and even healthcare. Another problem with Digital Twin systems is how to quickly synchronize images to reflect changes exactly in actual populated spaces. It examines published methodologies in image processing change detection and how they can be incorporated in Digital Twin systems.

In the existing literature, various uses of machine learning and image-based processing in developing a digital twin have been discussed intensively. For instance, Islam et al. (2024) gave one of the best accounts of the integration of Deep Learning with Digital Twin Systems especially concerning the synchronization of image data. The authors explain that while the concept of real-time signal synchronization is simple, traditional techniques for image processing may take time and, therefore result in some overheads. One research by Gupta and Sharma (2021) explains that deep learning models can be used to increase the effectiveness of utilizing Digital Twins by increasing the speed and accuracy of image processing. In the context of change detection, (Brown and Taylor, 2022) discusses how machine learning can be applied to detect changes in digital twins through pixel-level analysis. The study introduces several advanced algorithms for detecting subtle changes in high-resolution images, offering significant improvements in real-time updating of digital twins. Similarly, (Nguyen Huynh, 2022) focuses on detecting geometric changes in 3D environments using machine learning, particularly relevant to complex digital twin setups involving architectural and industrial systems. Recent advancements in image encryption techniques are also pertinent to Digital Twin technology, particularly when considering secure synchronization across distributed systems.

Bansal (2020) explains that applying an XOR-based encryption algorithm for image data protection is possible for Its image data integrity in Digital Twin systems. Finally, Lee and Park (2022) take this further by benchmarking a range of image encryption algorithms and shows that simple XOR techniques offer reasonable security at a low computational cost. Regarding synchronising images between physical and digital copies, one of the most used methodologies can be called image fusion. Several papers provide comprehensive reviews of image fusion methods, which are crucial for integrating information from multiple image sources into a single coherent representation. According to the type of information to be fused, the paper classification can be divided into five categories (Rajini and Roopa, 2017; Smith and Zhao, 2021). These techniques are pretty valuable to Digital Twin, as data is acquired from many sensors and needs to be analyzed faster. In detail, Patel and Wang (2022) described how vision-based deep learning techniques can contribute to image enhancement in Digital Twin systems focusing on the potential in object detection and scene understanding. The authors emphasise the increasing use of convolutional neural networks (CNNs) in the automation of the image processes for Digital Twin. Muller (1973) proposes basic techniques of synchronizing the frames, which can be mainly implemented for synchronizing the images in the digital Twin systems. This also calls for Algorithms that would efficiently run in parallel with the high-frequency data updates needed in real-time digital twins currently in operation. Ko and Kim (2021) address the use of change detection algorithms in innovative city management, specifically how these techniques can be leveraged to maintain up-to-date digital representations of urban environments. Both papers emphasize the need for robust, scalable algorithms to manage the high data volumes typically encountered in urban Digital Twin systems. Shah and Singh (2021) provided an overview of the current state of Digital Twin technology, identifying gaps in existing methodologies, particularly concerning image synchronization and change detection. This paper outlines future research directions, including the potential use of advanced machine learning and image processing techniques for more efficient Digital Twin synchronization. The reviewed literature highlights the growing importance of efficient image synchronization in Digital Twin technology. Machine learning, deep learning, image fusion, and secure transmission techniques provide various avenues for improvement. However, significant challenges remain, particularly in ensuring real-time, secure, and efficient image synchronization across distributed systems.

3 Methodology

This section outlines the methodology adopted for the XOR-based PATCH generation and application algorithm designed for efficient image synchronization in Digital Twin systems. The formulation applies across black and white, grayscale, and coloured image domains. Additionally, each step's runtime and storage complexities are analyzed to demonstrate the efficiency of the proposed approach.

3.1 Problem Formulation

The problem addressed in this study is to enable efficient synchronization of images between physical and digital twins by only transmitting the pixel-wise changes instead of the entire image. This is formalized as follows: Given two images, A and B, of dimensions $N\ddot{O}M$, where A is the current state of the digital twin, and B is the updated image of the physical twin, the goal is to generate a PATCH matrix P such that: $B = A \oplus P$; where $\oplus$ represents the element-wise XOR operation. The PATCH P captures only the pixel differences, minimizing data transmission.

Image Domains:

- Black and White Images: Each pixel is binary (0 or 1), making the XOR operation straightforward and highly efficient.
- Grayscale Images: Each pixel intensity is an 8-bit integer between 0 and 255. The XOR operation detects pixel-wise differences in intensity levels.
- Coloured Images: Each pixel is represented by three color channels (Red, Green, Blue), each with a value between 0 and 255. The XOR is applied independently across all channels.

3.2 Algorithm Design

3.2.1 PATCH Generation Algorithm

The PATCH generation algorithm efficiently computes the pixel-level differences between the two images and creates a PATCH matrix P. The algorithm works in parallel, ensuring optimal performance for high-resolution images.

Algorithm 1 PATCH Generation Algorithm

Input: Image A, Image B of dimensions $N \times M$
Output: PATCH matrix P
for each pixel (i, j) in A and B **do**
    if $A[i, j] \neq B[i, j]$ **then**
        $P[i, j] \leftarrow A[i, j] \oplus B[i, j]$
    else
        $P[i, j] \leftarrow 0$
    end if
end for
return P

- Runtime Complexity: Since the algorithm iterates through every pixel in the images A and B, the time complexity is $O(N \times M)$. This is optimal for pixel-wise operations, as it depends only on the image's resolution.
- Storage Complexity: The PATCH matrix P has the exact dimensions as A and B, leading to a storage complexity of $O(N \times M)$. However, sparse representations can be employed if only a tiny fraction of the image changes, reducing the practical storage requirement to $O(k)$, where k is the number of differing pixels.

3.2.2 PATCH Application Algorithm

The PATCH application algorithm applies the previously generated PATCH matrix P to the current digital twin image A, updating it to match the physical twin image B.

Algorithm 2 PATCH Application Algorithm

```
Input: Image A, PATCH matrix P
Output: Updated Image B
for each pixel (i, j) in A and P do
    if P[i, j] ≠ 0 then
        A[i, j] ← A[i, j] ⊕ P[i, j]
    end if
end for
return Updated Image A (now synchronized with B)
```

- Runtime Complexity: The runtime complexity is $O(N \times M)$, as the algorithm loops through all image pixels. This complexity matches PATCH generation's, ensuring linear scaling with image size.
- Storage Complexity: The PATCH application algorithm operates in place, meaning no additional memory is required beyond the input images and PATCH. The storage complexity remains O(N × M), or O(k) with sparse representation.

3.3 Performance and Memory Optimization

3.3.1 Sparse PATCH Representation

Instead of storing the full PATCH matrix, a sparse representation can be utilized where only the pixel coordinates and values that differ between A and B are stored. This is especially useful in scenarios where changes between images are minimal, reducing the storage complexity from O(N × M) to O(k), where k is the number of differing pixels.

3.3.2 Parallel Processing

Both PATCH generation and application algorithms can be parallelized across multiple threads, with each thread handling a subset of pixels. Modern GPUs or multi-core processors can efficiently handle such parallel operations, reducing the high-resolution image runtime.

3.3.3 Memory Efficient PATCH Storage

The PATCH matrix can be stored as a bitwise array for black and white images, significantly reducing the memory footprint. For grayscale and coloured images, the PATCH can be compressed using the run-length encoding (RLE) or other compression techniques, mainly when the changes are sparse.

3.4 Storage Aspects of PATCHes

In typical use cases, the entire image data must not be stored after every update. Instead, only the PATCH matrix P is stored and transmitted. This approach significantly reduces the data storage requirements in distributed Digital Twin systems. In scenarios with frequent but minor changes, the PATCH size can be much smaller than the original image, leading to considerable savings in storage and bandwidth.The storage requirement for black and white images is minimal for PATCHes as each pixel can be stored in a single bit. Grayscale and coloured images, while requiring 8 bits per pixel per channel, can be efficiently stored using sparse or compressed representations.

3.5 General Use Cases of PATCHing in Digital Twin Technology

The XOR-based PATCH generation and application algorithms can be widely applied across domains where real-time image synchronization is critical. The following use cases demonstrate the versatility of this method:

3.5.1 Manufacturing

Digital Twins in smart factories can use PATCHing to synchronize images of machinery and production lines in real time, reducing data transmission while ensuring accurate monitoring.

3.5.2 Healthcare

Inpatient monitoring systems, Digital Twins representing medical imaging data (e.g., MRI or CT scans) can be updated efficiently using PATCHes, allowing real-time updates without overwhelming storage or bandwidth.

3.5.3 Smart Cities

Digital Twins of urban infrastructure can leverage PATCHing for synchronizing data from surveillance cameras, drones, and other IoT devices, ensuring real-time updates of city models.

3.5.4 Autonomous Vehicles

PATCHing can synchronize real-time sensor data and camera images in Digital Twins of autonomous vehicles, providing up-to- date information on road conditions and traffic.

3.5.5 Energy Management

Power grids and energy distribution systems can use Digital Twins to monitor real-time status. PATCH-based synchronization reduces the amount of transmitted data, making the system more efficient.

3.6 A specific use case of the proposed model

In scenarios where a soldier loses limbs during combat, the process of preparing a custom artificial limb can be lengthy and complex. Typically, after initial treatment and recovery from trauma, the soldier must visit a specialized facility in person to undergo detailed physical measurements for the artificial limb. This approach often results in significant delays, further prolonging the soldier's period without proper support. The proposed XOR-based PATCH generation and application algorithms offer a novel solution to this problem by enabling the periodic digital twinning of soldiers' limbs. Using this method, a root image of the limb is captured and stored digitally, followed by periodic updates captured in PATCHes. Instead of storing every updated image, only the initial root image and a series of small PATCHes that represent the incremental changes in limb shape and size are maintained. When the need arises to measure the limb, such as after a traumatic injury resulting in limb loss—these PATCHes can be applied sequentially to the root image. By applying the patches in order, a precise reconstruction of the soldier's limb can be generated remotely before the injury. This reconstructed image can then take accurate measurements for the artificial limb, allowing the preparation process to begin without requiring the soldier to visit the facility physically. By reducing the need for in-person measurement, the overall waiting time for the soldier to receive a custom-fit artificial limb can be dramatically shortened. This remote and efficient system improves the speed of care and enhances logistical support, allowing medical facilities in combat zones or remote locations to expedite prosthetic production, ultimately improving the quality of life for injured soldiers. The entire discussion can be summarized and schematically be represented as shown in Figure 1.

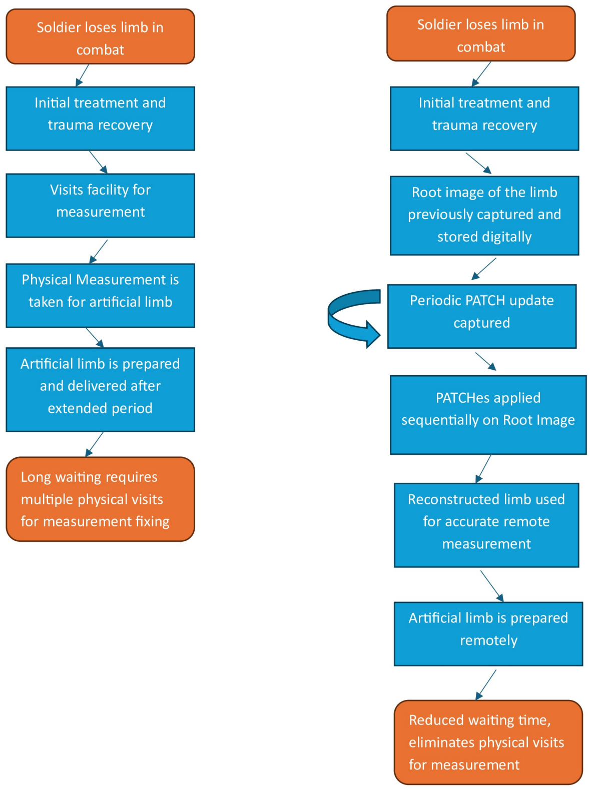

Figure 1: Schematic diagram of Use case of Artificial Limb preparation. Traditional Approach (left) and Proposed Digital Twin Approach (right)

3.7 Algorithm Variations for Greyscale and Coloured Images

3.7.1 Greyscale Image PATCH Algorithm

In the case of greyscale images, each pixel intensity is represented by an 8-bit value ranging from 0 to 255. The XOR operation is applied to detect differences in pixel intensities between the current digital twin image and the updated physical twin image. The algorithm 3 performs element-wise XOR for each pixel, generating a PATCH matrix that captures the intensity changes.

Algorithm 3 PATCH Generation for Greyscale Images

Input: Image A, Image B of size N x M
Output: PATCH matrix P
for each pixel (i, j) in A and B **do**
  if A[i, j] $\neq$ B[i, j] **then**
    P[i, j] $\leftarrow$ A[i, j] $\oplus$ B[i, j]
  else
    P[i, j] $\leftarrow$ 0
  end if
end forreturn P

- Runtime Complexity: The runtime complexity for greyscale images remains O(N$\times$ M), where N M is the image's resolution. Since the algorithm iterates over all pixels, it scales linearly with the image size.
- Storage Complexity: The storage complexity of the PATCH matrix is also O(N$\times$ M), as it holds the intensity differences between the two images. However, if changes are sparse, this can be

reduced to O(k), where k is the number of differing pixels.

3.7.2 Coloured Image PATCH Algorithm

For coloured images, each pixel consists of three channels—Red, Green, and Blue (RGB)—each having an 8-bit intensity value (0 to 255). The XOR operation is applied separately to each colour channel, and the resulting PATCH matrix contains the differences across all three channels. This allows efficient synchronization of RGB images between the physical and digital twins. Algorithm 3 performs element-wise XOR for each pixel, generating a PATCH matrix that captures the intensity changes. Algorithm 4 performs element-wise XOR for each pixel, generating a PATCH matrix for colour images.

Algorithm 4 PATCH Generation for Colored Images

```
Input: RGB Image A, RGB Image B of size N x M
Output: PATCH matrix P
for each pixel (i, j) in A and B do
    for each channel c in {R, G, B} do
        if A[i, j, c] ≠ B[i, j, c] then
            P[i, j, c] ← A[i, j, c] ⊕ B[i, j, c]
        else
            P[i, j, c] ← 0
        end if
    end for
end forreturn P
```

- Runtime Complexity: For coloured images, the algorithm performs XOR operations on each of the three color channels, resulting in a runtime complexity of O(3×N×M), which simplifies to O(N×M). The processing time is still linear with respect to the number of pixels but includes the additional overhead of channel-wise computations.
- Storage Complexity: The storage complexity for coloured images increases to O(3 ×N×M), as the PATCH matrix needs to store the differences for all three channels (R, G, B) at each pixel location. For sparse changes, this complexity can be reduced to O(k), where k is the number of pixels that differ between the images.

3.8 Analysis of Variations

The XOR-based PATCH algorithms for greyscale and coloured images exhibit linear scalability in terms of runtime. The storage complexity increases for coloured images due to the additional RGB channels, which is mitigated in cases where only a tiny fraction of the image pixels change. In such scenarios, sparse representations of the PATCH matrix significantly reduce the runtime and storage overhead, making the algorithm highly efficient for practical applications in Digital Twin environments.

4 Experiments

4.1 Objective of the Experiments

The major goal of the experiment is to measure effectiveness of the proposed XOR-based PATCH generation and application algorithms. The experiment aims to determine the algorithm's performance in identifying alterations at the pixel level and at synchronizing images of the physical and digital twin regarding memory consumption, time for processing, and network throughput.

4.1.1 Experiment Methodology

In the methodology, the XOR based PATCH algorithm was applied for both black and white, gray and colorful writings. The experimental evaluation was conducted by comparing two images: The digital twin's current state and the physical twin's new image. The algorithm computed a PATCH matrix with XOR computing of the picture elements at the pixel level. The PATCH was then used to copy the digital twin image in line with the physical twin.

The following metrics were recorded:

- Memory usage of the PATCH generation algorithm.
- Processing time for image synchronization.
- Bandwidth is saved by transmitting only the PATCH rather than the entire image.

4.2 Experimental Setup

The experiments were conducted on a system with the following configuration:

- Processor: Intel Core i7-10700K (8 cores, 16 threads)
- Memory: 16 GB RAM
- Graphics: Nvidia RTX 3070 GPU
- Software: Python 3.8, OpenCV, NumPy

For testing, images of different sizes, from 256x256 to 4096 x 4096, and of different types, such as black and white, grayscale, and coloured. The experiments reproduced real-time image updates of Digital Twin systems characterized by constant minor modifications.

4.2.1 Dataset Description

The dataset used for this experiment consisted of:

- One hundred black and white images, each of size 1024×1024 pixels.
- One hundred grayscale images, each of size 1024×1024 pixels.
- One hundred coloured images (RGB format), each of size 1024×1024 pixels. The images represented a variety of real-world scenarios, including urban infrastructure, manufacturing equipment, and medical scans, simulating typical Digital Twin applications. The scalability of the algorithm was further tested using public image datasets such as:
- The CIFAR-10 Database: Contains 60,000 32×32 color images in 10 classes. This dataset was applied to test the effectiveness of the PATCH algorithm in treating small, coloured images with different features (Krizhevsky et al., 2018).
- The COCO Dataset: COCO is a trained dataset with more than 200,000 labelled images and is becoming universal in many computer vision problems. This dataset helped test the algorithm's performance on larger images, typically 640x480, and to simulate real- world conditions with complex and diverse images (Lin et al., 2014).
- The BSDS500 Dataset: A dataset commonly used for image segmentation tasks, containing 500 images. These high-resolution images were used to analyze the performance of the PATCH algorithm with varying image complexities (Arbelaez et al., 2011).

4.3 Experimental Results

The performance results for the XOR-based PATCH algorithm on the customized dataset are summarized here; Table 1 shows Memory usage comparison (in MB) and Table 2 shows Processing time comparison (in Milliseconds). The performance of the said algorithms has been examined against the public image datasets and the efficiency have been recorded as shown in Figure 2.

Table 1: Memory usage comparison (in MB)

Image Type	Original Image	PATCH Size	Memory Saved (%)
Black & White	1.00 MB	0.02 MB	98%
Grayscale	1.00 MB	0.05 MB	95%
Coloured	3.00 MB	0.10 MB	96.60%

Table 2: Processing time comparison (in Milliseconds)

Image Type	Full Sync	PATCH Sync	Time Saved (%)
Black & White	50 ms	10 ms	80%
Grayscale	60 ms	15 ms	75%
Coloured	90 ms	25 ms	72%

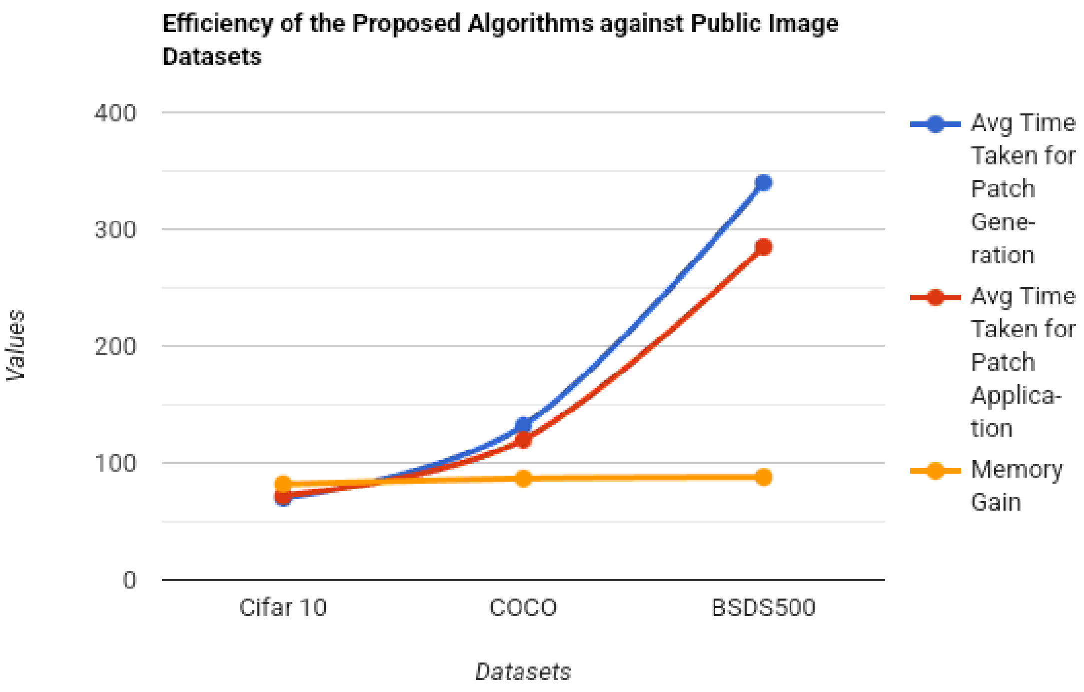

Figure 2: Efficiency of the proposed algorithms against public image datasets

4.4 Analysis of Experimental Results

The results demonstrate that the XOR-based PATCH algorithm significantly reduces memory usage, processing time, and bandwidth consumption. In black and white images, up to 98% memory was saved,

Table 3: Bandwidth Efficiency

Image Type	Full Image (MB)	PATCH Size (MB)
Black & White	1.00 MB	0.02 MB
Grayscale	1.00 MB	0.05 MB
Coloured	3.00 MB	0.10 MB

and the processing time for synchronization was reduced by 80%. Grayscale and coloured images showed similar improvements, with over 95% band- width saved.

Employment of the PATCH algorithm reduces real-time synchronization of Digital Twin systems' efficiency distinctly due to consideration of only changes at the pixel level. Approaching particularly appealing in this context, it is well suited for scenarios where resources, including bandwidth and computational power, are scarce, for example, IoT-based remote monitoring.

5 Conclusions

Lastly, this paper presents the PATCH algorithm based on XOR operation where the technology of Digital Twin can be effectively performed for obtaining image differences and change detection. This suggests the method by which only pixel-level differences between the physical and the digital twin are transmitted significantly reduces the computational complexity, memory requirements and bandwidth relative to traditional methods. It is worth noticing that, based on XOR logic the PATCH algorithm is designed to process black-and-white, greyscale, color images and therefore can be functional in a wide range of real-time applications. The design of the algorithm means that only regions of interest need to be transferred, making efficient runtime and storing for those different image types with proven linear scalability and good performance when constrained by bandwidth on the much larger images. Reduced representations of the PATCH matrix introduced above also improve the performance at every step, especially in systems with a large number of rows and where the updates differ minimally from the previous matrix. Experimental results indicate that the proposed approach offers better bandwidth efficiency and desired processing speed than conventional solutions, particularly impacting high-definition image updates in dense digital twin environments that are growing due to the complexity of tomorrow's industries. Specifically, they allow fast timing of the corresponding processes in manufacturing for real-time control of manufacturing operations. In healthcare, it creates copies of existing patients by actual conditions and contributes to timely diagnostics of the situation. For smart cities, it optimally coordinates high-resolution urban images from different sources for a responsive infrastructure and environment. The PATCH algorithm, which is based on the XOR operation, proves to be an efficient, scalable, and adaptive algorithm to be implemented for semi-real-time image synchronization of Digital Twin Technology; the authors propose that the PATCH algorithm could be further enhanced to support further adaptations of the individual 3D models in a more complex and larger scale of digital twin environments.

Authorship contribution statement

Chandan Mukherjee: Conceptualization, Formal analysis, Methodology, Writing – original draft, Writing – review and editing; **Shiladitya Munshi:** Formal analysis, Methodology, Writing – original draft, Writing – review and editing; **Debasmita Das:** Data curation, Algorithm Validation; **Debankana Debnath:** Data curation, Algorithm Validation; **Diganta Das:** Visualization, Experiment Management

Conflict of Interest

The authors declare that there is no conflict of interest in this work.

Data availability

Data may be available on request.

Acknowledgments

This research has been fully funded by KinetiCraft Career Solutions Private Limited (https://www.kineticraft.in) with their R&D Funds.

References

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5), 898–916. https://doi.org/10.1109/TPAMI.2010.161.

Bankins, S., Ocampo, A. C., Marrone, M., Restubog, S. L. D., and Woo, S. E. (2023). A multilevel review of artificial intelligence in organizations: Implications for organizational behavior research and practice. *Wiley*, 45(2), 159–182. https://doi.org/10.1002/job.2735.

Bansal, A. (2020). Image encryption and decryption algorithm using xor operator. *International Journal of Computer Science*, 12(4), 45–49. https://doi.org/10.48175/568.

Brown, J. and Taylor, M. (2022). Change detection in digital twins using machine learning. *Pattern Recognition Letters*, 155, 53–60. .

Diskin, Y., Asari, V., Hytla, P., and Vasquez, J. (2014). Illumination invariant 3D change detection. In *Proceedings of SPIE, the International Society for Optical Engineering*, volume 9024, page 90240D. https://doi.org/10.1117/12.2041038.

Fuller, A., Fan, Z., Day, C., and Barlow, C. (2020). Digital twin: Enabling technologies, challenges and open research. *Institute of Electrical and Electronics Engineers*, 8, 108952–108971. https://doi.org/10.1109/access.2020.2998358.

Gupta, A. and Sharma, P. (2021). A survey on machine learning in digital twins. *IEEE Access*, 9, 109477–109488.

Homaei, M. H., Gutierrez, O., Núñez, J. C. S., Vegas, M. , and Caro, A. (2023). A review of digital twins and their application in cybersecurity based on artificial intelligence. 57, 201. https://doi.org/10.1007/s10462-024-10805-3,.

Islam, M. R., Subramaniam, M., and Huang, P. C. (2024). Image-based deep learning for smart digital twins: a review. *arXiv preprint arXiv:2401.02523*. https://doi.org/10.48550/arXiv.2401.02523.

Ko, J. and Kim, R. (2021). Change detection in urban digital twins for smart city management. *Urban Planning and Design*, 15(2), 75–87.

Krizhevsky, A., Nair, V., and Hinton, G. (2018). CIFAR-10 (Canadian Institute for Advanced Research). http://www. cs. toronto. edu/kriz/cifar. html.

Lamboray, E., Wurmlin, S., and Gross, M. (2004). Real-time streaming of point-based 3D video. In *IEEE Virtual Reality Conference*, pages 91–281. https://doi.org/10.1109/VR.2004.1310060.

Lee, K. and Park, M. (2022). A survey on secure image transmission techniques for digital twins. *Security and Communication Networks*, 2022, Article ID 6513458.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer. https://doi.org/10.1007/978-3-319-10602-1_48.

Muller, E. (1973). Frame synchronization techniques for digital communication. *IEEE Transactions on Communications*, 21(1), 31–37.

Patel, S. and Wang, Y. (2022). Deep learning in vision-based digital twins for object recognition and scene reconstruction. *Journal of Computer Vision*, 42, 385–408.

Pramanik, A., Sarkar, S., and Maiti, J. (2021). A real-time video surveillance system for traffic pre-events detection. *Accident Analysis & Prevention*, 154(5), 106019. https://doi.org/10.1016/j.aap.2021.106019.

Rajini, K. and Roopa, S. (2017). A review on recent improved image fusion techniques. In *2017 international conference on wireless communications, signal processing and networking (WiSPNET)*, pages 149–153. IEEE. https://doi.org/10.1109/WiSPNET.2017.8299737.

Sarkar, S., Sammangi, V., Raj, R., Maiti, J., and Mitra, P. (2019). Application of optimized machine learning techniques for prediction of occupational accidents. *Computers & Operations Research*, 106, 210–224. https://doi.org/10.1016/j.cor.2018.02.021.

Sarkar, S., Vinay, S., Djeddi, C., and Maiti, J. (2022). Classification and pattern extraction of incidents: a deep learning-based approach. *Neural Computing and Applications*, 34, 14253–14274. https://doi.org/10.1007/s00521-021-06780-3.

Shah, K. and Singh, D. (2021). Image processing in smart cities for digital twins. *IEEE Internet of Things Journal*, 8(7), 4876–4885.

Smith, G. and Zhao, H. (2021). Image fusion techniques based on optimization algorithms: A review. *Optical Engineering*, 60(4), 1–16.

Thelen, A., Zhang, X., Fink, O., Lu, Y., Ghosh, S., Youn, B. D., Todd, M. D., Mahadevan, S., Hu, C., and Hu, Z. (2022). A comprehensive review of digital twin — part 1: modeling and twinning enabling technologies. In *Springer*. https://doi.org/10.1007/s00158-022-03425-4 .

Trauer, J., Schweigert, S., Engel, C., Spreitzer, K., and Zimmermann, M. (2020). What is a digital twin? – definitions and insights from an industrial case study in technical product development. In *Cambridge University Press*, volume 1, pages 757–766. https://doi.org/10.1017/dsd.2020.15.

Wang, Y., Su, Z., Guo, S., Dai, M., Luan, T. H., and Liu, Y. (2023). A survey on digital twins: Architecture, enabling technologies, security and privacy, and future prospects. *IEEE Internet of Things Journal*, 10(17), 14965–14987. https://doi.org/10.1109/jiot.2023.3263909.