International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18 (2026) pp. 1-15

https://doi.org/10.70917/ijcisim-2026-0112
Article

Research on the Technical Path to Optimize the Reconstruction of Large-Scale Historical Scenes in Historical Documentaries Using Parallel Distributed Computing Techniques

Hexi Wang and Mingjie Wang *

School of Art and Media, Beijing Normal University, Beijing, 100091, China; wmjjmw12345@126.com

Abstract: The continuous development of distributed computing technology provides a new path for high-precision reconstruction of historical scenes. In this paper, we propose a technical optimization framework based on parallel distributed computing for the problems of low efficiency and lack of clarity in the reconstruction of large-scale historical scenes in historical documentaries. By integrating digital elevation model (DEM) and 3D Gaussian sputtering (3DGS) methods, combined with the DAG scheduling mechanism of Apache Spark framework and multi-factor weighted resilient distributed dataset (RDD) caching strategy, the reconstruction speed and rendering quality of the scene are significantly improved. The experiments show that the parallel computing nodes are set to 15, the nodes adopt an interval of 1500m, and the number of single file computations is 11 times to obtain higher modeling efficiency. The ambiguity of scene reconstruction with the introduction of parallel distributed computing is reduced to less than 17%, and the average ambiguity is lower than 16%. The values of three evaluation indexes are better than those of the comparison algorithms.

Keywords: parallel distributed computing; DEM; 3DGS; RDD cache optimization; historical scene reconstruction

1. Introduction

In recent years, the most significant change in documentary narrative techniques has been the emergence of the narrative method of reconstructing historical events at their original sites [1]. We can observe that the parts being “reconstructed” are all aspects of the past that cannot be directly reenacted. Such historical events are referred to by television producers as “narrative gaps” [2]. Creators reenact these events on screen through methods such as reenactments, bringing these authentic historical events back to life before the audience, thereby serving to bridge these narrative gaps [3-5]. This creative form, which combines documentary segments with fictional segments, allows documentaries that cannot capture the original historical appearance to tell a complete and smooth narrative. In this sense, it can be said to restore the original historical appearance, and has therefore been widely adopted in documentary production [6-9]. To enhance the emotional impact of documentaries and make them more plot-driven and dramatic, many new storytelling techniques commonly used in feature films have been introduced into documentary production [10-11]. Especially in the reconstruction of historical events, innovative techniques have emerged, with the most typical being the new trend of storytelling in both content and form [12].

However, due to changes in the audience demographics of documentaries in today's society, viewers, who have been influenced by Hollywood blockbusters, are no longer satisfied with simple visual effects [13]. The viewership ratings of television documentaries are not only influenced by content but also significantly impacted by the quality of visual imagery [14]. In other words, when reconstructing historical events in documentaries, greater attention must be paid to form and methodology. It is essential to employ superior technical skills and vivid forms to engage viewers' emotions, ensuring the content

aligns with contemporary viewing habits [15-18]. Through the reconstruction of historical scenes and the portrayal of characters' emotions, viewers can authentically experience the humanity depicted in the documentary, thereby allowing the true historical stories in the documentary to deeply resonate in viewers' understanding of humanity and society [19-21].

Optimization paths for large-scale scene reconstruction have now been widely applied across various fields. Literature [22] explores the optimization of large-scale, real-time 3D scene reconstruction on mobile devices. Considering that large-scale dense reconstruction methods require significant memory and high-performance GPU computing, sparse reconstruction methods can be employed to conserve computational and memory resources. Literature [23] utilizes dynamic spatial hash truncated symbol distance fields for real-time 3D scene reconstruction. Based on this technology, very large scenes can be reconstructed and rendered in real time without GPU computing, facilitating user interaction activities. Literature [24] designs an advanced method for optimizing 3DGS (Three-dimensional Gaussian splatting) for high-resolution, large-scale scenes. By employing a pruning mechanism to reduce the excessive diffusion of Gaussian distributions, it achieves high rendering fidelity with reduced computational and storage costs. Literature [25] investigates optimization techniques for the rapid reconstruction of scenes such as buildings and streets, utilizing CUDA-based stereo matching algorithms to optimize embedded computing devices, thereby effectively improving the efficiency and quality of scene reconstruction. However, there is limited research on the application of scene reconstruction optimization techniques in the creation of historical documentaries, making it an area of significant research value.

This paper focuses on the innovative application of parallel distributed computing technology in historical terrain reconstruction. The DEM model is constructed based on historical topographic maps to restore ancient geomorphic features, and combined with the GIS platform to visualize the 3D scene and highlight the special terrain. Introduce 3DGS method to enhance rendering realism and support dynamic scene expression through anchor point mechanism. Design Spark's DAG stage division strategy to optimize task parallelism and accelerate the terrain data processing pipeline. Multi-dimensional innovative RDD caching strategy to reduce data communication load and data recalculation overhead.

2. Historical Scene Reconstruction and Optimization Based on Parallel Distributed Computing

2.1. DEM Modeling Principles and Methods

Digital elevation model (DEM) is a digital representation and simulation of continuously distributed topographic features on the earth's surface by means of discretely distributed elevation data, which is an ordered collection describing the spatial location of surface units and the distribution of topographic attributes, and is a spatial distribution model representing the actual topographic features.As a kind of spatial data, DEM, with each point possessing the real three-dimensional coordinates, facilitates three-dimensional visualization and statistical analysis, and is an effective method to study the surface process and construct geomorphology. The specific steps of the modeling method are as follows: firstly, extract the terrain elevation and attribute information from the historical topographic maps, try to improve the above topographic information by combining the relevant data, and restore the historical topographic features; then use the relevant mathematical algorithms and GIS technology to extract the elevation data information of the historical topography, set the parameters of thinning and sampling, etc., and construct the DEM; and then, through the electronic sand table technology, achieve the 3-D reconstruction of the historical topographic maps containing the features of the geomorphology. The reconstruction of historical geomorphological map is then realized through electronic sand table technology.

2.2. 3D Scene Reconstruction

Three-dimensional reconstruction of historical landscapes, that is, the use of GIS technology to simulate geographic phenomena that have been lost or blurred, to maximize the original appearance, and on the basis of the analysis and discussion, to examine the historical process in different spatial scales. In recent decades, due to the influence of human development, the disc-shaped depressions in Basin B are gradually blurred and disappeared. In order to reconstruct this geographic phenomenon, we need to select the research materials that have less influence on the disc-shaped depressions by human activities and are more accurate as the basic data. Shunzhi topographic maps were published before the large-scale development and construction of China in the modern era, and they were also the first topographic maps published in China using modern surveying techniques and in accordance with formal geodetic norms. Using this historical point in time to construct the 3D scene map can better reflect the original remains of the saucer-shaped depressions geomorphology in Basin B a century ago.

Combined with GIS technology, this paper utilizes MapGIS as a platform to construct the 3D scene

with 1:25000 straight topographic map data as the basic information, referring to the same period of aerial and satellite films and related data information. Figure 1 is the three-dimensional reconstruction technology process.

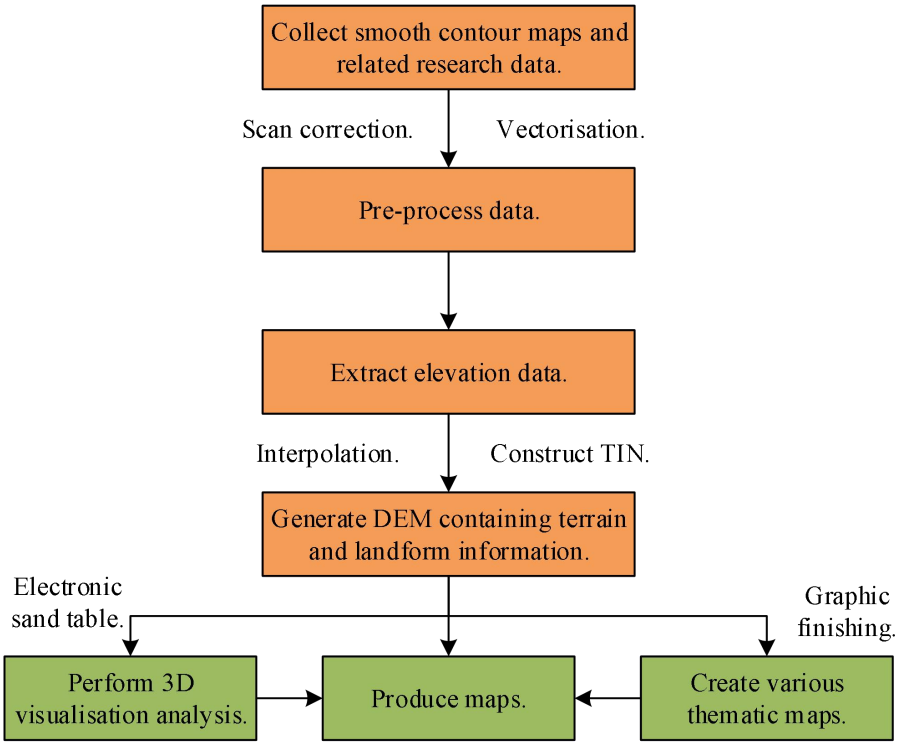

Figure 1. Three-dimensional reconstruction technology process.

2.2.1. Data Pre-Processing

The purpose of data preprocessing is to construct the DEM of the research target with elevation data that have spatial coordinate system and elevation attributes.Firstly, the vectorized data scanned from the straight topographic map are corrected for errors and assigned with elevation, and the elevation field should be of double-precision type, so as to make it have spatial coordinate system and elevation attributes; then, the layers of modern and man-made terrains are eliminated, and only the elements that can represent the original natural topography are retained, and the most important ones are the contour lines and elevation data; finally, with reference to the aerial and satellite film and on-site exploration data and related information, the blank areas of the data are repaired, and the obtained elevation data are systematically checked, and the invalid data are modified and eliminated, so as to make them satisfy the data format requirements of the MapGIS platform.

2.2.2. 3D Scene Creation

1) Establishment of DEM. The key to the production of 3D scene is to establish DEM, the establishment of DEM firstly needs to select DEM model and meshing method, set parameters, etc. There are more models and meshing methods for DEM, and MapGIS platform provides a variety of models and meshing methods. Due to the complex topography and special morphology of the saucer-shaped depressions in the B basin, the DEM constructed should be able to reflect the fine features of this special topography, therefore, this paper adopts the TIN model and the pan kriging method to construct the DEM.

The modeling process also needs to set other parameters such as sparse sampling, grid spacing, search method and number of grid lines. Whether these parameters are set reasonably or not will also affect the accuracy of the model, and the reasonable parameters need to be set flexibly depending on the characteristics of the research data, and generally need to be repeatedly tested based on the accuracy of the model to adjust the model parameters, in order to complete the modeling in a real sense, and lay a good foundation for the later research.

After solving the above problems, you can start modeling, the basic steps of modeling are: open the MapGIS platform, select the DTM analysis module under the spatial analysis subsystem, open the elevation point/line file, and perform "line data elevation point extraction" under the menu "Process Point Line"; Then use "Quickly Generate Triangulation Network" and "Organize Triangulation Network" under the menu "TIN Model" to obtain the TIN file; Finally, process the TIN file in the "Discrete Data Meshing" option under the menu GRD Model. The grid method is Pan Kriging method, the grid line is set to 250, and the search type is quadripartite search. After the parameters are set, clicking Confirm will output a GRD file, so that the DEM modeling is basically completed.

2) Three-dimensional scene generation. The three-dimensional scene is mainly to build DEM to show the terrain undulation, according to the different elevation of the different color values, can produce three-dimensional topographic maps, to express the different changes in the terrain undulation, MapGIS platform through the three-dimensional window under the DTM analysis module and the electronic sand table module to generate the three-dimensional scene. The electronic sand table module is based on DEM, through the three-dimensional interactive terrain visualization environment, with roaming browsing, terrain analysis, data overlay and other functions, this paper adopts the electronic sand table module to generate three-dimensional scenes. According to the needs of observation and research, the scene display mode, effect and observation angle and other parameters are set and adjusted to enhance the degree of realism of the three-dimensional scene drawing; and then export the three-dimensional three-dimensional stereo map through the “Output three-dimensional scene” command in the file menu. The three-dimensional scene map of disc-shaped depressions in Basin B generated by this method can clearly and intuitively show the individual characteristics of the disc-shaped depressions in Basin B, which are similar to discs in flat (profile) form, and the individual characteristics of the surrounding uplift and the ring-shaped residual mound form, as well as the group characteristics of disc-shaped depressions that are widely distributed and in large quantities.

The research results of the GIS reconstruction of the three-dimensional scene can be summarized, and it can be concluded that the evolution process of the saucer-shaped depressions in Basin B. Basin B has been formed since the Holocene period under the combined effect of both natural and man-made factors. A unique geomorphological phenomenon, the disk-shaped depression, has been formed through the evolution and construction of various factors such as surface water erosion and artificial modification over a long period of time. Most of these special depressions were destroyed by surface water erosion, and waterlogged depressions were formed in the lower terrain, which gradually evolved into human settlements in the area, and the highlands between the riverbeds were less preserved due to the influence of surface water on the disc-shaped depressions and their group geomorphology. In recent years, especially with the rapid development of China's national economy, people's efforts to transform and demand for nature have been strengthened as never before, which has had a great adverse impact on the ecological environment of Basin B. The water sources have been reduced significantly, the environment has been seriously polluted, and the wetland areas and lakes have shrunk considerably, which has led to a drastic reduction or even the danger of the disappearance of the disk-shaped depressions, and they are now distributed in the highland areas between the rivers in the way of group existence, and are only found in the highland areas between the rivers. The disc-shaped depressions in Basin B are not only a living history and teaching material for the study of geomorphological development and evolution of Plain A and Basin B, but also have important effects on the landscape, economy and society of the basin, which should be highly valued and protected.

The three-dimensional scene map generated by using the above method contains significantly more effective information than the two-dimensional plan view, and can fully display the detailed changes of the saucer-shaped depressions in Basin B, with a strong sense of three-dimensionality and hierarchy, and provide a more graphic and intuitive display of the geomorphology of the study area, which makes it easy to synthesize all kinds of effective geographic information for analysis and research. This method also has good application prospects in 3D visualization analysis and production of various thematic maps.

2.3. Three-Dimensional Gaussian Splash (3DGS) Methods

The 3DGS method represents the scene using a set of anisotropic 3D Gaussians, which inherits the differential properties of the body representation for efficient scene rendering through a rasterized sputtering process. Each point initialized from structure-from-motion (SFM) is designated as the center position of the 3D Gaussian u , and each 3D Gaussian can be expressed as

$$G(x) = e^{\frac{-1}{2}(x-u)^T \Sigma^{-1}(x-u)} \tag{1}$$

where x denotes any sampling position in 3D space; Σ denotes the covariance matrix of the 3D

Gaussian, Eq.

$$\Sigma = RSS^T R^T \tag{2}$$

In this paper, we maintain the positive semidefiniteness of the covariance matrix Σ by scaling the matrix S and rotating the matrix R. Unlike traditional body representations, the 3DGS method renders the scene by rasterized sputtering rather than resource-intensive ray tracing. Denoting a pixel point on a 2D image by p, the point-based rendering method obtains the color of the pixel on the image

$$C(p) = \sum_{i \in N} T_i \alpha_i c_i \tag{3}$$

where T_i denotes the cumulative transparency of the i th 3D Gaussian on the ray, Eq.

$$T_i = \prod_{j=1}^{i-1} \left(1 - \alpha_j\right) \tag{4}$$

α_i denotes the opacity value of the i th 3D Gaussian projection into the 2D image plane at pixel point p, with equation

$$\alpha_i = \sigma_i e^{\frac{-1}{2}(p-u_i)^T \Sigma^{'} (p-u_i)} \tag{5}$$

σ_i denotes the opacity of each 3D Gaussian, u_i denotes the UV coordinates of each 3D Gaussian projected onto the 2D image plane; c_i denotes the color of each 3D Gaussian.

The 3D Gaussian is only suitable for representing static scenes, and applying a time-conditional learnable parameter to each 3D Gaussian results in a loss of motion spatio-temporal continuity. In order to enable the 3D Gaussian to represent dynamic scenes, this paper dynamically guides the distribution of local 3D Gaussians through anchor points according to the real structure of dynamic scenes, resulting in a hierarchical and region-aware scene representation.

2.4. DAG Construction and Staging

In the open source distributed computing system (Apache Spark), the extraction and computational processing of preprocessed historical terrain elevation data using parallel distributed computing techniques can improve the scene rendering fidelity of 3DGS methods, thus enhancing the realism of large-scale historical scene reconstruction in historical documentaries.

In Apache Spark, Directed Acyclic Graph (DAG) is a key concept used to represent the Spark task execution flow, which describes the dependencies between resilient distributed datasets (RDDs) and the transformation operations on these RDDs. Figure 2 shows the DAG flow of a Spark job.

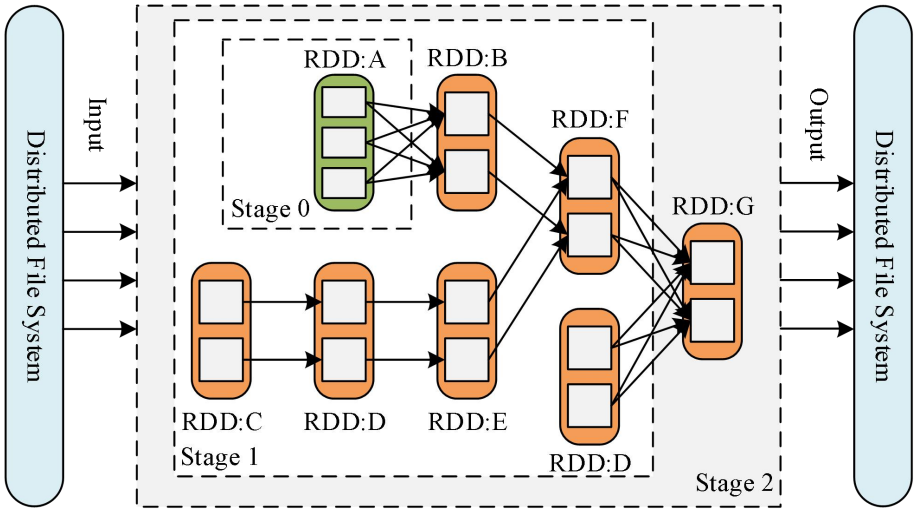

Figure 2. The DAG of Spark jobs.

The DAG construction process involves a series of transformations and optimization processes from user code to logical and physical execution to final task execution.

First, the user defines a series of RDD transformation operations and action operations by writing the Spark application, which constitute the logical execution plan of the Spark application. The transformation operations and action operations in the code will be parsed into logical operations in Spark, and the logical execution constitutes the directed acyclic graph (DAG) of the job. Each node represents a transformation operation of an RDD and edges represent data dependencies. The physical execution plan is divided into phases, and a phase contains a set of tasks that can be executed in parallel. There are dependencies between phases and tasks within the same phase can be executed in parallel. The DAG scheduler then sorts the phases according to their dependencies and constructs the execution order of the entire DAG and submits it to the cluster scheduler. The cluster scheduler assigns the tasks to the executors in the cluster for execution, and the executors execute the tasks in the cluster. Finally, the executor returns the results to the driver or writes them to the external storage system.

Overall, the DAG build process ensures that Spark applications execute on the cluster, improving overall performance.Spark's DAG execution engine is one of the foundations of its high performance.

2.5. Important Factors for RDD Caching

In the process of completing large-scale historical scene reconstruction and rendering fidelity enhancement, it involves the processing of a large amount of data, and effective caching of such data is related to the improvement of modeling computation speed and system memory consumption.

Least Recently Used (LRU) is a common cache elimination strategy, which relies only on access order for cache management, easily replacing frequently used hot data out of the cache, with low memory utilization and costly job recomputation. Cache management should consider multiple factors at the same time, such as the access frequency, size, and correlation between data, etc. A comprehensive consideration of multiple factors is required to evaluate the importance of data. By quantifying the factors affecting cache, it is possible to more accurately determine which data should be kept in memory and which data can be eliminated. It can improve the cache hit rate, reduce unnecessary data recalculation, and improve the execution efficiency of jobs. In this paper, we synthesize the main factors of RDD cache value are the stage bloodline length of RDDs, the frequency of use, the number of partitions, and the complexity of arithmetic.

A developer can submit one or more contexts in a cluster. During the execution of an application, it is divided into Job jobs by the Action action operator, and the jobs are divided into phases based on wide and narrow dependencies, and there are multiple RDDs in each phase.If the job directed acyclic graph contains i phases, the context can be denoted as. If a stage contains j RDDs, it denotes the entire dataset of the stage. If the number of starting partitions of a stage is k partitions, the RDD can be represented, where represents the partition.

In context, there are multiple partitions for the stage being divided and a partition corresponds to only one pipeline (or pipeline). The $Line_{ik}$ denotes the k th pipeline of the i th stage, then $Line_{ik} = \{SP_{i1k}, SP_{i2k}, \ldots, SP_{ijk}\}$ can be denoted as the whole pipeline, and the computation time of the pipeline is the sum of the computation time of all the partitions it contains, i.e.:

$$T_{Line_{ik}} = \sum_{x=1}^{j} T_{SP_{ixk}} \tag{6}$$

Denoting the total number of pipelines in a particular stage by $Lines_i$, $Lines_i = \{Line_{i1}, Line_{i2}, \ldots, Line_{ik}\}$ is denoted as the total number of pipelines in that stage, and the computation time of a particular stage is the maximum value of the time overhead of all pipelines in that stage, i.e.:

$$T_{Stage_i} = \max\left(T_{Line_{i1}}, T_{Line_{i2}}, \ldots, T_{Line_{ik}}\right) \tag{7}$$

Then, the execution time of the whole job can be expressed as:

$$T_{Job} = \sum_{y=1}^{i} T_{Stage_y} \tag{8}$$

1) Stage bloodline length (linelongRDD)

The stage linelength is expressed as the linelength of the RDD in the current stage. The lineage is the information of the parent RDD on which the computation process of the RDD depends. The bloodline records the source and transformation process of the data, which is the key for Spark to realize data processing and recovery. During the computation process of RDD, if the data of a partition is lost or the computation fails, Spark can recompute the data of that partition based on the blood relation of RDD. Therefore, bloodline is crucial for RDD distributed computation to ensure the correctness of computation. In cache replacement, the longer the stage bloodline is, the greater the impact on the cache. When the data in the cache is replaced out, a longer stage bloodline length for RDD means that the amount of data that needs to be recomputed will be more, resulting in a higher cost of data recovery. RDDs with longer stage bloodline lengths require more computational overhead to recalculate the data because their dependencies are more complex and the impact of failures will have a wider range, increasing the computational overhead and resource consumption of the job. Figure 3 shows the Spark stage dependencies.

RDDs in Spark can only be derived from their parent RDDs by the transformation operator, so their distance to the direct parent RDD is 1.0, denoted by $lineLong$. By default all RDDs have only one parent dependency, then the stage blood maximum can be expressed as:

$$\max StageLong = \sum_{o}^{j} lineLong \tag{9}$$

If RDDA is the l th in the stage, then the stage blood length linelongRDD of RDD is equal to l and satisfies the following equation:

$$1 <= l <= \max StageLong \tag{10}$$

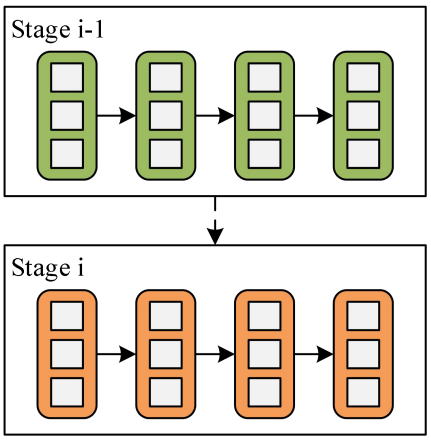

Figure 3. Spark stage dependency relationship.

2) RDD reuse frequency (frequentRDD)

RDD reuse frequency indicates the number of times an RDD is referenced in the same job.Spark job execution mechanism adopts lazy lazy loading, before the application submits the TaskSet to the Executor, it will construct a directed acyclic graph based on the dependencies of RDDs. In a directed acyclic graph, caching the RDDs used at high frequency reduces the overhead during computation. Because in the case of repeated use of the same dataset, caching of RDD data can avoid repeated computation and increase the speed of computation. Frequent reading of data from disk is avoided by reducing disk IO operations. Figure 4 shows the DAG for a Pagerank job.Links are called multiple times during the job iteration. If the RDD is referenced N times, the reuse frequency of the RDD can be expressed as:

$$frequentRDD = N \tag{11}$$

It is easy to know that during the execution of SC, RDDs are computed according to the dependency

relationships, and each partition needs to obtain data and compute processing from the partitions whose parents depend on RDDs. If $Farthers_{ijk}$ is the set of dependent parent partitions of partition SP_{ijk}, the partition computation cost is the sum of data read-in and processing, i.e.:

$$T_{SP_{ijk}} = input(Farthers_{ijk}) + cal(Fathers_{ijk}) \tag{12}$$

In a Spark job, a set of logically identical data is processed in parallel by multiple executors. Since all partitions of the RDD have the same commit time, the RDD computational cost is the maximum of the computational cost among all the partitions, i.e:

$$T_{RDD_{ij}} = \max\left(T_{SP_{ij1}}, T_{SP_{ij2}}, \ldots, T_{SP_{ijk}}\right) \tag{13}$$

If the data of the j th RDD in the i th stage is lost, then the recalculation cost of that RDD and the actual execution time of the job can be expressed as:

$$CT_{RDD_{ij}} = \sum_{1}^{i-1} T_{Stage} + \sum_{1}^{j} T_{RDD} \tag{14}$$

$$T'_{Job} = T_{Job} + (N-1) * CT_{RDD_{ij}} \tag{15}$$

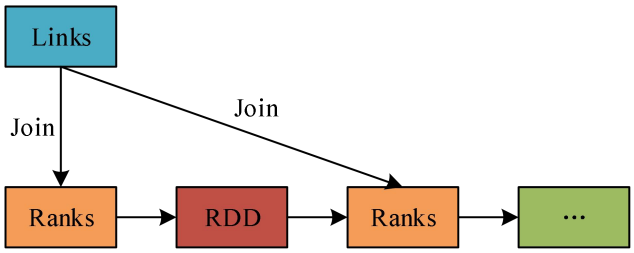

Figure 4. The DAG of the Pagerank assignment.

It follows that the loss or replacement of hotspot data out of the cache can significantly increase the execution time of a job. If multiple operations are performed on the same data, these operations can be accelerated by caching RDDs that are used at high frequency.

3) RDD operator complexity (transRDD)

RDD arithmetic complexity indicates the number of parent RDDs referenced by the creation of the RDD. Initial RDDs can be created by parallelization, local files or HDFS files. In addition to the initial RDD, each RDD has at least one parent RDD. i.e., the parent RDD has a one-to-one, or many-to-one relationship with the child RDDs. Figure 5 shows the RDD transformation relationship. If RDDC is generated by RDDA by directly executing the transformation operator, such as the map operator, then RDDA and RDDC have a one-to-one relationship. Similarly, RDDD is jointly generated by RDDA and RDDB, such as join operator, then there is a many-to-one relationship between RDDA, RDDB and RDDD.

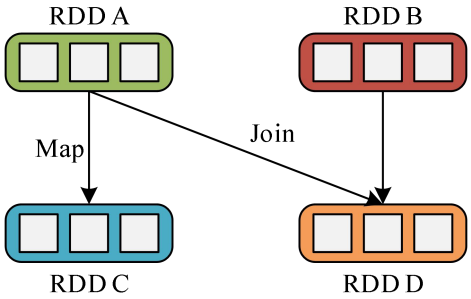

Figure 5. RDD transformation relationship.

Since the RDD is generated by the parent RDD by performing the transformation operator, the number of parent RDDs determines the operator complexity of the RDD. If the RDD is transformed from

S parent RDDs, then the arithmetic complexity of the RDD can be expressed as:

$$transRDD = S \tag{16}$$

4) Number of partitions (partRDD)

Spark assigns partitions to Executor for computation during scheduling, and the Executor assigns a Task to each partition for processing. When the server is down, each partition is reassigned a Task to recover, and the more partitions there are, the more resources are consumed for recovery. Figure 6 shows the number of RDD partitions.The number of partitions does not change when RDDE generates RDDF, the number of partitions is 3.0.But when RDDG is created by RDDE, the number of partitions changes from 3.0 partitions to 2.0 partitions. So, when RDD data is lost, the more the number of partitions, then the more nodes to recover the task, which seriously affects the overall progress of the job.

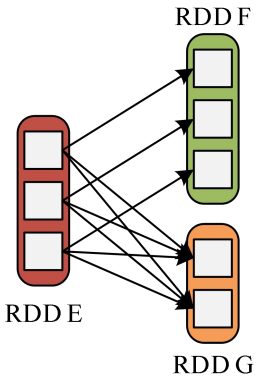

Figure 6. Number of RDD partitions.

Spark provides a corresponding API to change the number of partitions, the user only needs to call the $repartition(\ )$ method to change the number of partitions of the RDD, by setting the parameter to determine the number of modified partitions. If the parameter set is T , then the total number of partitions of the current RDD can be expressed as the variable:

$$partRDD = T \tag{17}$$

3. Historical Scene Reconstruction Practices under Parallel Distributed Computing

3.1. Determination of Optimal Computational Nodes and Function Quantities

3.1.1. Rendering Time Required for Different Number of Nodes

In this paper, by controlling the number of computational nodes in the Spark cluster, the computational time required by the parallel algorithm is counted when the number of nodes produces a change, so as to determine the number of parallel nodes with the highest computational efficiency and to optimize the speed of reconstruction of historical scenes. Figure 7 shows the impact of the number of parallel nodes on the computation time of the model pipeline. Table 1 shows the comparison of the computation time required for different numbers of computing nodes. When the number of parallel nodes is increased from 3 to 15, the model computation time is reduced from 69.358 min to 30.607 min, and when the nodes adopt an interval of 1500 m, the computation time required for 15 nodes is the least 30.607 min. based on this, this paper sets the number of computational nodes in the Spark cluster to 15, and each node adopts an interval of 1500m to obtain higher efficiency in modeling historical scenes.

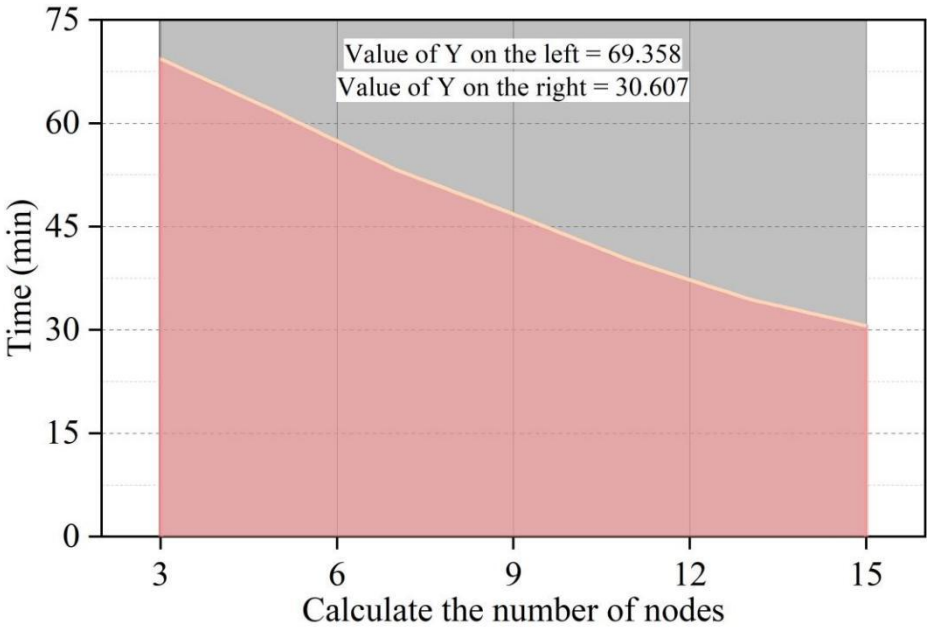

Figure 7. Influence of the number of parallel nodes on the model computing time.

Table 1. Computing time required for different numbers of computing nodes.

Number of nodes	Adopt intervals 1500m(min)	Adopt intervals 1200m(min)	Adopt intervals 900m(min)	Adopt intervals 600m(min)
3	69.358	70.442	72.471	75.251
5	61.553	62.631	64.627	68.414
7	53.281	57.364	58.399	62.186
9	46.833	47.914	49.949	54.739
11	40.066	44.147	45.185	48.972
13	34.509	38.592	40.622	43.406
15	30.607	31.619	34.654	38.443

3.1.2 Comparison of the Number of Stores and Communication Load for Different Number of Computations

The number of stored elevation data files is related to the number of computations for each input file. In this section, a comparison experiment is set up to compare the number of computations for each input file between the 2 methods, Least Recently Used (LRU) caching strategy and RDD caching strategy, to determine the number of stored files and communication load of the different methods. Figure 8 shows the number of files stored for the 2 methods. Fig. 9 shows the communication load of the 2 methods. In the process of increasing the number of calculations for each file from 2 to 14, the total file cache of LRU surges from 10.235*1035 to 19.327*1035, meanwhile, the communication load rate increases from 50.961% to 98.742%, which occupies more memory, leading to a heavier computational burden, which is not conducive to the rapid construction of the historical scene model. As for the RDD caching strategy, with the increasing number of individual files, the total piece caching volume slowly increases from 10.093*1035 to 10.554*1035, which is only an increase of 0.461*1035; the communication load rate increases from 50.071% to 62.981%, which is only an increase of 12.910%. Compared with LRU, RDD caching strategy can better process the file data. Meanwhile, from the cache changes, it can be seen that when the number of calculations is about to reach 12 times, the total file caches of the 2 caching methods show a significant increase, so the maximum number of calculations of a single file is set to 11 times as a way to reduce the amount of cache for parallel distributed computation and to improve the computational efficiency.

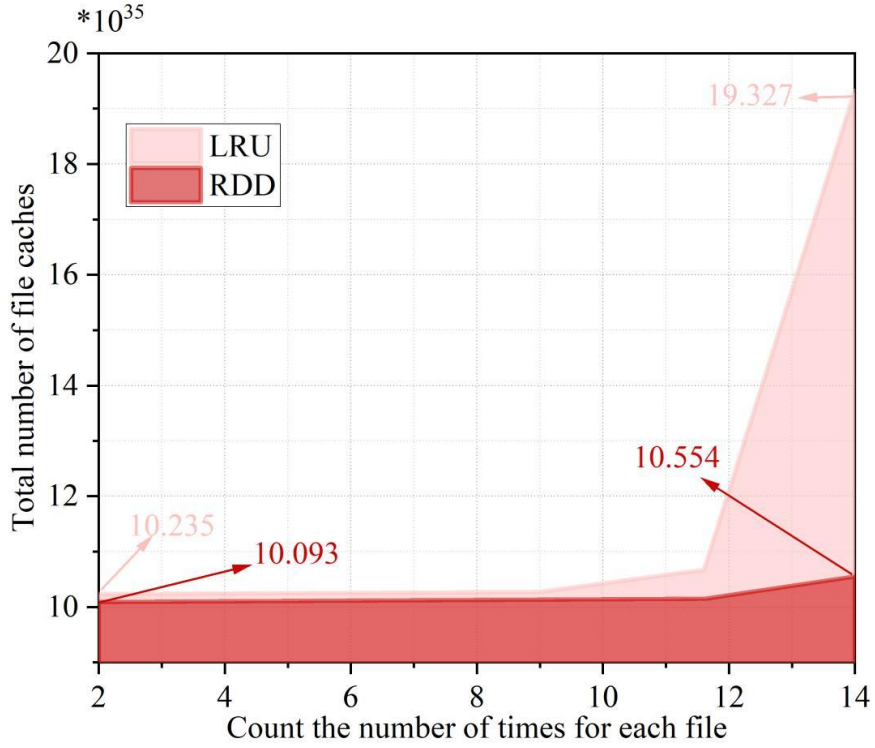

Figure 8. Ratio of the file storage quantities of the two methods.

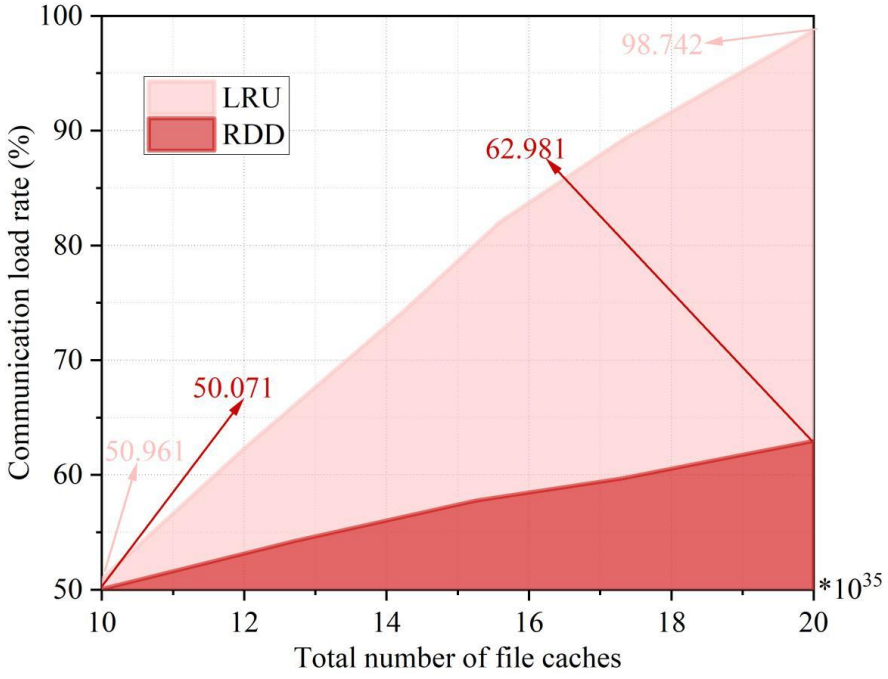

Figure 9. Ratio of communication loads of the two methods.

3.2. Comparison of Scene Reconstruction Effects

3.2.1 Comparison of Reconstructed Scene Ambiguity Before and After Parallel Distributed Computing

In order to verify the advantages of introducing parallel distributed computing technology in this paper, a large-scale scene 1 from a historical documentary related to the saucer-shaped depressions in a certain B basin is selected as the object of reconstruction study (scene 1 contains 10 important topographic features), and comparative experiments are set up. The blurriness of the reconstructed scene before and after the introduction of parallel distributed computing is analyzed. Figure 10 shows the comparison results of the blurriness of the reconstructed scenes before and after parallel distributed computing. The reconstruction fuzzy degree of the 10 feature points before the introduction of parallel distributed computing reaches 40.67%, 45.98%, 43.21%, 42.83%, 42.79%, 45.02%, 46.17%, 47.23%, 45.71%, and 47.14%, which is more than 40%, indicating that the reconstruction fuzzy degree of this historical scene is higher before the introduction of parallel distributed computing that a clear reconstruction could not be realized. And after the introduction of parallel distributed computing, the reconstruction fuzzy degree of feature points decreased to 10.24%, 15.51%, 12.78%, 12.04%, 12.36%, 14.59%, 15.71%, 16.23%, 15.28%, 16.71%, which are all lower than 17%, and the fuzzy degree has decreased by nearly 23%. It can be judged that the introduction of parallel distributed computing technology can improve the clarity of historical scene modeling.

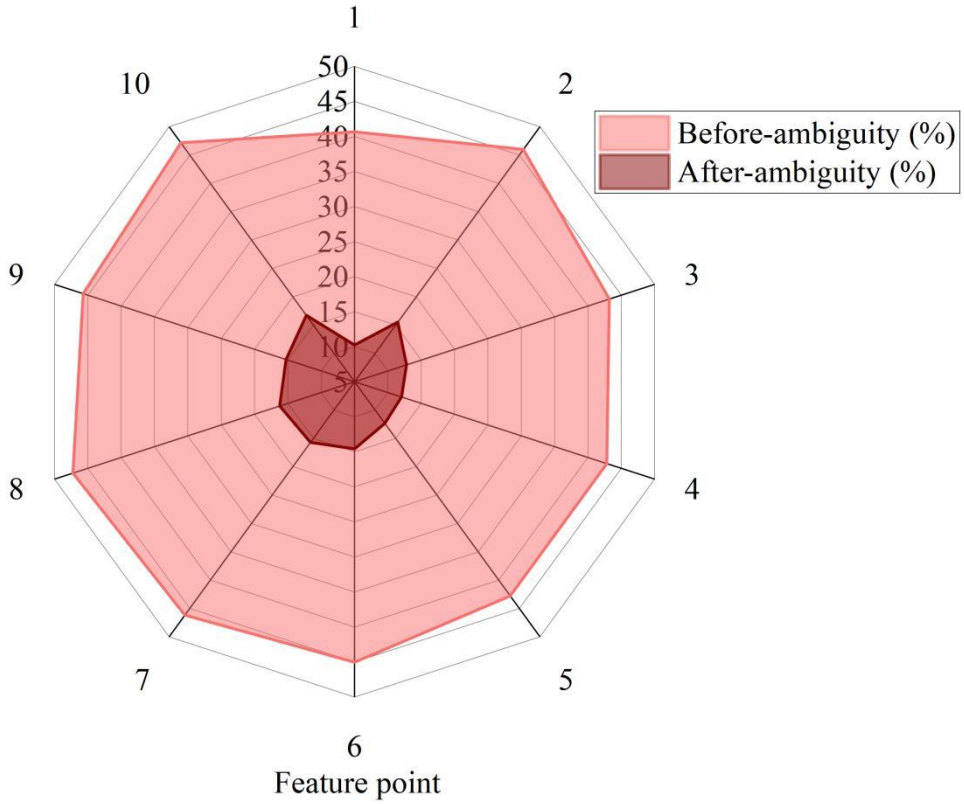

Figure 10. Reconstruct the contrast of scene blurriness.

3.2.2 Mean Fuzzy Analysis of Multiple Reconstructions

Fig. 11 shows the change of average blurring degree of scene 1 reconstructed 15 times after the introduction of parallel distributed computing (taking the average value of 10 feature points). During the 15 reconstructions, the average blurring degree of scene 1 is as low as 10.98% and as high as 15.67%, none of which exceeds 16%. Through several reconstruction experiments, the optimization effect of historical scene reconstruction after the introduction of parallel distributed computing is tested.

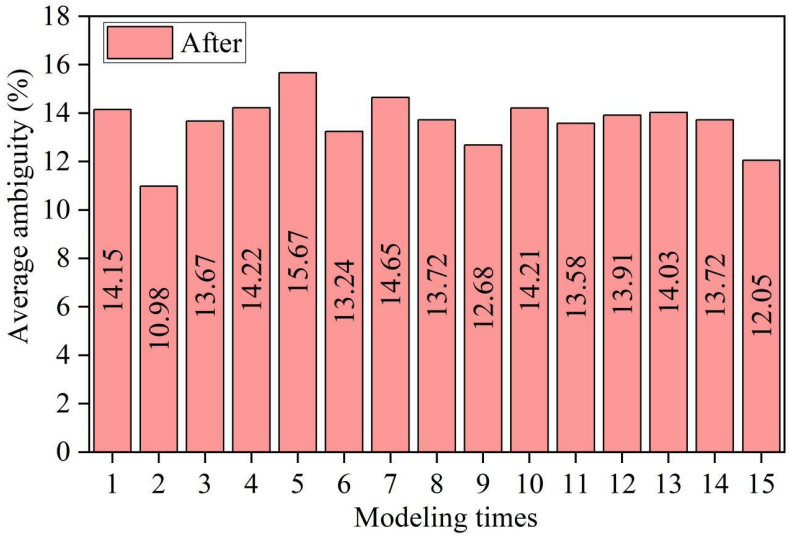

Figure 11. Reconstruct the average ambiguity variation 15 times.

3.3. Comparison of the Effect of Different Algorithms for Historical Scene Reconstruction

The NeRF and NeRF-W algorithms are chosen as comparison algorithms to compare the overall historical scene reconstruction effect with the parallel distributed computing method in this paper. After completing the training of the algorithms, the three methods were utilized to complete the reconstruction of three large-scale scenes in a historical documentary related to a B-basin saucer depression, respectively. The historical scene reconstruction effect was evaluated by three metrics: peak signal-to-noise ratio (PSNR), multi-scale structural similarity (MS-SSIM), and image similarity (LPIPS). Table 2 shows the comparison of historical scene reconstruction effects of different methods. In the reconstruction of three historical scenes, the PSNR of this paper's method reaches 40.754, 43.976, and 40.773, which is the only PSNR value over 40 among the three methods.The MS-SSIM is 0.891, 0.853, and 0.815, which is higher than the comparison methods. And in the LPIPS index, the LPIPS values of this paper's method are the smallest: 0.316, 0.202, and 0.374.Therefore, in the 3D reconstruction task of a large-scale historical scene, the introduction of parallel distributed computation can reduce the data cache share, optimize the computational efficiency, and improve the reconstruction quality.

Table 2. Comparison results of historical scene reconstruction effects.

Scene number	Evaluation index	NeRF	NeRF-W	Ours
A	PSNR	30.811	35.073	40.754
	MS-SSIM	0.725	0.784	0.891
	LPIPS	0.502	0.673	0.316
B	PSNR	38.742	39.106	43.976
	MS-SSIM	0.792	0.798	0.853
	LPIPS	0.491	0.534	0.202
C	PSNR	37.920	38.596	40.773
	MS-SSIM	0.784	0.736	0.815
	LPIPS	0.571	0.566	0.374

4. Conclusion

In this paper, an optimization path for large-scale historical scene reconstruction based on parallel distributed computing is proposed and verified. The results show that: 1) When the number of nodes in the Spark cluster is increased to 15, the nodes are spaced at 1500 m, and a single file can be computed up

to 11 times, the scene reconstruction efficiency is higher and the data communication loss is less. 2) The ambiguity of the feature points of the 10 reconstruction scenes is 10.24%, 15.51%, 12.78%, 12.04%, 12.36%, 14.59%, 15.71%, 16.23%, 15.28% and 16.71%, which is much lower than before the introduction. 3) In the evaluation of the three large-scale historical scene reconstruction indicators, the PSNR index exceeded 40, the MS-SSIM was greater than 0.8, and the LPIPS was the lowest 0.202, which was better than the comparison algorithm.

Future work can further explore the acceleration potential of different computing architectures for real-time rendering, and improve the realism of large-scale historical scene reconstruction.

References

1. Aaltonen, J., & Kortti, J. (2015). From evidence to re-enactment: history, television and documentary film. Journal of Media Practice, 16(2), 108-125.
2. Honess Roe, A. (2017). Interjections and Connections: The Critical Potential of Animated Segments in Live Action Documentary. Animation, 12(3), 272-286.
3. Wu, P., Liu, D., Ma, J., Gu, L., & Tong, J. (2019). Reconstructing the man-made Yellow River flood of Kaifeng City in 1642 AD using documentary sources. International Journal of Disaster Risk Reduction, 41, 101289.
4. Ellis, J. (2021). How documentaries mark themselves out from fiction: a genre-based approach. Studies in Documentary Film, 15(2), 140-150.
5. Majeed, T. (2019). Animated documentary: A dramatic art of documentary storytelling. International Journal of Communication and Media Studies, 9(1), 1-12.
6. Chen, S., Su, Y., Fang, X., & He, J. (2020). Climate records in ancient Chinese diaries and their application in historical climate reconstruction–a case study of Yunshan Diary. Climate of the Past Discussions, 2020, 1-28.
7. Brusaporci, S., Ruggieri, G., Sicuranza, F., & Maiezza, P. (2017, November). Augmented reality for historical storytelling. The INCIPICT project for the reconstruction of tangible and intangible image of L'Aquila historical centre. In Proceedings (Vol. 60, No. 1, p. 1083). MDPI.
8. Terrone, E. (2020). Documentaries, Docudramas, and Perceptual Beliefs: Terrone Documentaries, Docudramas, and Perceptual Beliefs. The Journal of Aesthetics and Art Criticism, 78(1), 43-56.
9. Lobovikov-Katz, A. (2019). 3D OBJECT RECONSTRUCTION IN A PRE-DIGITAL ERA: CASE STUDY IN THE HISTORY OF RESTORATION. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 42, 735-739.
10. Kerrigan, S., & Velikovsky, J. T. (2016). Examining documentary transmedia narratives through The Living History of Fort Scratchley project. Convergence, 22(3), 250-268.
11. Podara, A., Giomelakis, D., Nicolaou, C., Matsiola, M., & Kotsakis, R. (2021). Digital storytelling in cultural heritage: Audience engagement in the interactive documentary new life. Sustainability, 13(3), 1193.
12. Saxton, L. (2020). A true story: defining accuracy and authenticity in historical fiction. Rethinking History, 24(2), 127-144.
13. Robinson, L. (2015). Chinese documentary as public culture. International Communication of Chinese Culture, 2, 69-75.
14. Käser, S. (2017). Documentary Image Sequences. Visible Language.
15. Ferdani, D., Fanini, B., Piccioli, M. C., Carboni, F., & Vigliarolo, P. (2020). 3D reconstruction and validation of historical background for immersive VR applications and games: The case study of the Forum of Augustus in Rome. Journal of Cultural Heritage, 43, 129-143.
16. Shin, D., Ren, Z., Sudderth, E. B., & Fowlkes, C. C. (2019). 3d scene reconstruction with multi-layer depth and epipolar transformers. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 2172-2182).
17. Nguyen, V. T., Jung, K., Yoo, S., Kim, S., Park, S., & Currie, M. (2019, December). Civil War battlefield experience: Historical event simulation using augmented reality technology. In 2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR) (pp. 294-2943). IEEE.
18. Yi, Y. J., Rahim, M. H. A., & Sannusi, S. N. (2016). Techniques application on cultural and artistic documentary production: A study of simon schama's power of the art. Jurnal Komunikasi: Malaysian Journal of Communication, 32(2), 489-524.
19. Urban, M. (2015). Documentary film as historical narrative. Ars Aeterna, 7(2), 31-43.
20. Ramírez, G. (2024). The March of Time: News Documentaries and the Dramaturgy of History. Film & History: An Interdisciplinary Journal, 54(2), 39-58.
21. Ruchel-Stockmans, K. (2021). From amateur video to new documentary formats: citizen journalism and a reconfiguring of historical knowledge. Media and mapping practices in the Middle East and North Africa, 139-158.
22. Dryanovski, I., Klingensmith, M., Srinivasa, S. S., & Xiao, J. (2017). Large-scale, real-time 3D scene reconstruction on a mobile device. Autonomous Robots, 41, 1423-1445.
23. Klingensmith, M., Dryanovski, I., Srinivasa, S. S., & Xiao, J. (2015, July). Chisel: Real Time Large Scale 3D Reconstruction Onboard a Mobile Device using Spatially Hashed Signed Distance Fields. In Robotics: science and systems (Vol. 4, No. 1).
24. Liu, W., Guan, T., Zhu, B., Xu, L., Song, Z., Li, D., ... & Yang, W. (2025). Efficientgs: Streamlining gaussian splatting for large-scale high-resolution scene representation. IEEE MultiMedia.

25. Jiang, J., Zou, G., & Qi, C. (2024). Accelerated Reconstruction of Scenes using CUDA-based Parallel Computing. IEEE Access.