ISSN: 2349-6002

Text And Depth Controllable Video Generation Using AI

Mrs. Nithya B[1], Deepika S[2], Sharmila S[3], Shamitha S[4]
[1]*HOD/CSE, Department of Artificial Intelligence and Data Science, Surya Group of Institutions, Anna University*
[2,3,4]*Department Of Artificial Intelligence and Data Science, Surya Group of Institutions, Anna University*
doi.org/10.64643/IJIRTV12I11-201013-459

***Abstract*—Text and depth guided controllable video generation is an advanced AI-based approach that synthesizes videos using textual descriptions and depth information. The system proposes an automated AI-driven solution that converts textual scripts into visually coherent videos with minimal user intervention.**

The system employs Natural Language Processing (NLP) techniques to analyze, preprocess, and segment user-provided scripts into meaningful scenes. These scenes are then transformed into visually relevant video segments using latent diffusion models pre-trained for still image synthesis and promoted for video generation through temporal modules.

***Index Terms*—Text-to-Video Generation, Depth Estimation, Latent Diffusion Models, Natural Language Processing, Temporal Consistency, AI Video Synthesis, Script Analysis, Scene Generation.**

I. INTRODUCTION

A. Problem Statement

Traditional video creation requires significant manual effort, skilled professionals, and expensive production resources, making the process time- consuming and inaccessible to non-experts.

Existing AI-based video tools lack depth control, resulting in spatially inconsistent and visually unrealistic outputs that fail to accurately represent scene geometry.

Generating temporally coherent, depth-aware videos from text alone remains an open and challenging problem in computer vision research. The absence of proper depth guidance leads to flickering frames, poor object placement, and unnatural motion sequences in generated videos.

B. Objectives

The primary objective of this work is to develop an AI-driven system capable of generating high- quality, depth-controllable videos from user- provided textual scripts. The system aims to integrate NLP-based script analysis with latent diffusion models to produce temporally consistent, spatially accurate, and visually coherent video content.

A secondary objective is to build a modular pipeline that includes scene segmentation, depth- guided video generation, voiceover synthesis, and a user-friendly web interface for seamless interaction. The system is designed to reduce computational overhead while maintaining high output quality for diverse use cases including media, education, and entertainment.

C. Organization of Paper

The remainder of this paper is organized as follows: Section II reviews related work on existing text-to-video generation techniques and their limitations. Section III covers system analysis including the existing system, its drawbacks, and the proposed approach with a feasibility study.

Section IV describes the system design covering architecture, modules, data flow, and database design, followed by Section V on implementation details. Section VI presents the results and discussion, and Section VII concludes the paper with suggestions for future work along with a list of references.

II. RELATED WORK

A. Traditional IDS Techniques

Early text-to-video systems such as Synthesia and VEED.IO relied on stock footage matching and AI avatars to transform scripts into videos without deep scene understanding or spatial modeling. Methods like Make-A-Video (Singer et al., 2023) extended text-to-image diffusion models to the video domain by leveraging temporal priors and large-scale training data.

Video Fusion (Luo et al., 2023) proposed decomposed diffusion models to separately handle temporal and spatial generation, improving frame quality and motion coherence. Control Video introduced conditional control mechanisms for text- guided video editing, enabling targeted transformations on existing video content using diffusion-based frameworks.

B. Limitations of Existing Research

Despite significant progress, existing methods predominantly rely on text prompts alone, lacking the ability to integrate structured depth information for precise spatial control. This results in spatially inconsistent videos where object distances and 3D scene layouts are inaccurately represented.

Most current models also suffer from high computational costs, requiring large-scale GPU clusters and extensive training data with text-video-depth alignment annotations. Temporal inconsistency across frames remains a persistent challenge, with many models producing flickering artifacts and unstable object boundaries in generated sequences.

III. SYSTEM ANALYSIS

A. Existing System

Existing AI-powered script-to-video platforms such as Synthesia, Steve.AI, VEED.IO, and Flixier automate the transformation of text into finished videos by leveraging stock media libraries, AI avatars, and automated subtitle generation. These platforms are widely adopted for creating explainer videos, training materials, promotional content, and social media posts with minimal manual effort.

While these tools provide accessible and quick video generation capabilities, they fundamentally lack deep integration of 3D depth information or spatially aware scene composition. The generated outputs are constrained by pre-existing media assets and AI avatar templates, limiting creative control and visual accuracy.

B. Drawbacks

The primary drawback of existing systems is their inability to generate depth-accurate videos, leading to unrealistic object placement and poor 3D spatial representation in the output. Temporal inconsistency is another critical issue, as frames are often generated independently, resulting in flickering, unstable objects, and discontinuous motion sequences.

High computational demands make these systems impractical for real-time or resource-constrained environments, while dependency on large labeled datasets limits scalability. Additionally, most platforms offer restricted customization and cannot handle complex multi-object scenes with accurate depth relationships between elements.

C. Proposed System

The proposed system integrates an NLP-based Script Analysis Engine with a depth-guided AI Content Generation pipeline to produce spatially consistent and visually coherent videos from raw textual input. The system employs a Scene and Asset Generation Engine that converts each script segment into a visually rich scene using generative AI models enhanced with depth conditioning.

A Voiceover Generator module converts script segments into emotionally appropriate high-quality speech synchronized with generated video scenes using advanced text-to-speech synthesis. The modular design ensures each component — NLP processing, depth estimation, video generation, and media composition — operates independently while maintaining seamless data flow across the pipeline.

D. Feasibility Study

The technical feasibility of the proposed system is supported by the availability of pre-trained latent diffusion models, mature NLP libraries, and powerful GPU computing resources that can handle the computational requirements of real-time video generation. Modern hardware configurations with NVIDIA RTX 30-series GPUs and 32GB RAM provide sufficient processing power to run the proposed architecture efficiently.

Economically, the system leverages open-source frameworks including Python, Flask, React.js, and FFmpeg, significantly reducing development costs while maintaining high output quality. Operationally, the web-based user interface ensures accessibility for non-technical users, making the system viable for deployment in educational, media production, and enterprise environments.

ISSN: 2349-6002

IV. SYSTEM DESIGN

A. Architecture

The system architecture follows a layered modular design comprising a frontend user interface layer, a backend control and routing layer, an NLP processing layer, an AI generation layer, and a media composition and storage layer. Data flows sequentially from script input through NLP analysis to AI-driven visual and audio generation, culminating in a fully composed output video.

The architecture integrates React.js for the frontend, Node.js and Express.js for backend API management, Python-based NLP engines, and latent diffusion models for scene synthesis with depth conditioning. Cloud storage services manage the persistence and delivery of generated assets, ensuring scalability and efficient file management across system components.

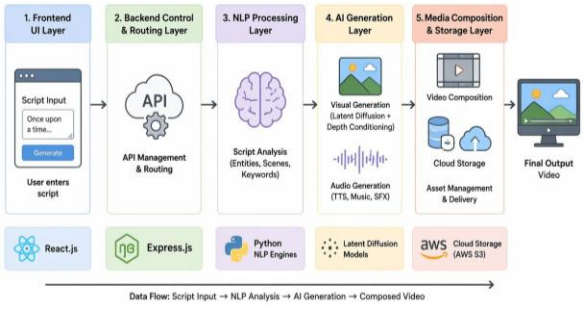

Figure 1: System Architecture

B. Module Description

The User Interface Module provides an interactive web-based frontend where users can input scripts, configure video generation parameters, preview outputs, and download final videos using HTML, CSS, and JavaScript. The Backend and Control Module, built on Node.js and Express.js, manages API orchestration, validates inputs, handles errors, and coordinates data flow between all system components.

The NLP and Script Processing Module analyzes input text to extract characters, actions, emotions, settings, and scene transitions using transformer- based language models and NLP libraries. The AI Content Generation Module leverages latent diffusion models with temporal extensions to convert scene prompts and depth maps into high- quality, temporally consistent video frames.

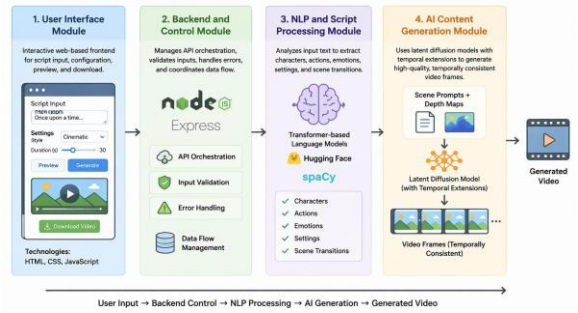

Figure 2: Module Description

C. Data Flow Diagram

The data flow begins with the user submitting a textual script through the web interface, which is then forwarded to the backend controller that triggers the NLP Script Processing Module for structured scene extraction. Extracted scene data is passed to the AI Content Generation Module, which produces image frames with depth-guided spatial conditioning, followed by audio synthesis from the Voiceover Generator.

The Media Processing and Video Composition Module receives generated frames and audio tracks, performs frame sequencing, synchronization, and video rendering using FFmpeg, and stores the final output in cloud storage. The composed video is then served back to the user through the web interface for preview and download, completing the end-to- end pipeline.

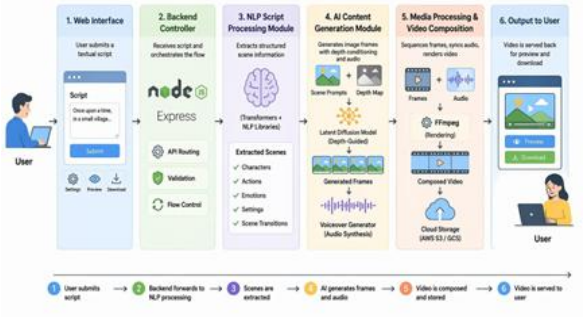

Figure 3: Data Flow Diagram DFD

D. Database Design

The database schema is designed to store user session data, script inputs, generated scene metadata, audio files, and final video outputs using a structured key-value and relational data model. Each user session is associated with a unique session ID that links to a sequence of generated scenes, audio clips, depth maps, and video frames for efficient retrieval and playback.

ISSN: 2349-6002

Generated media assets are stored in cloud-based file management systems with references maintained in the database to support efficient access, update, and deletion of associated files. The schema accommodates scalability by separating metadata storage from binary media asset storage, ensuring fast query performance even as the number of stored video projects grows.

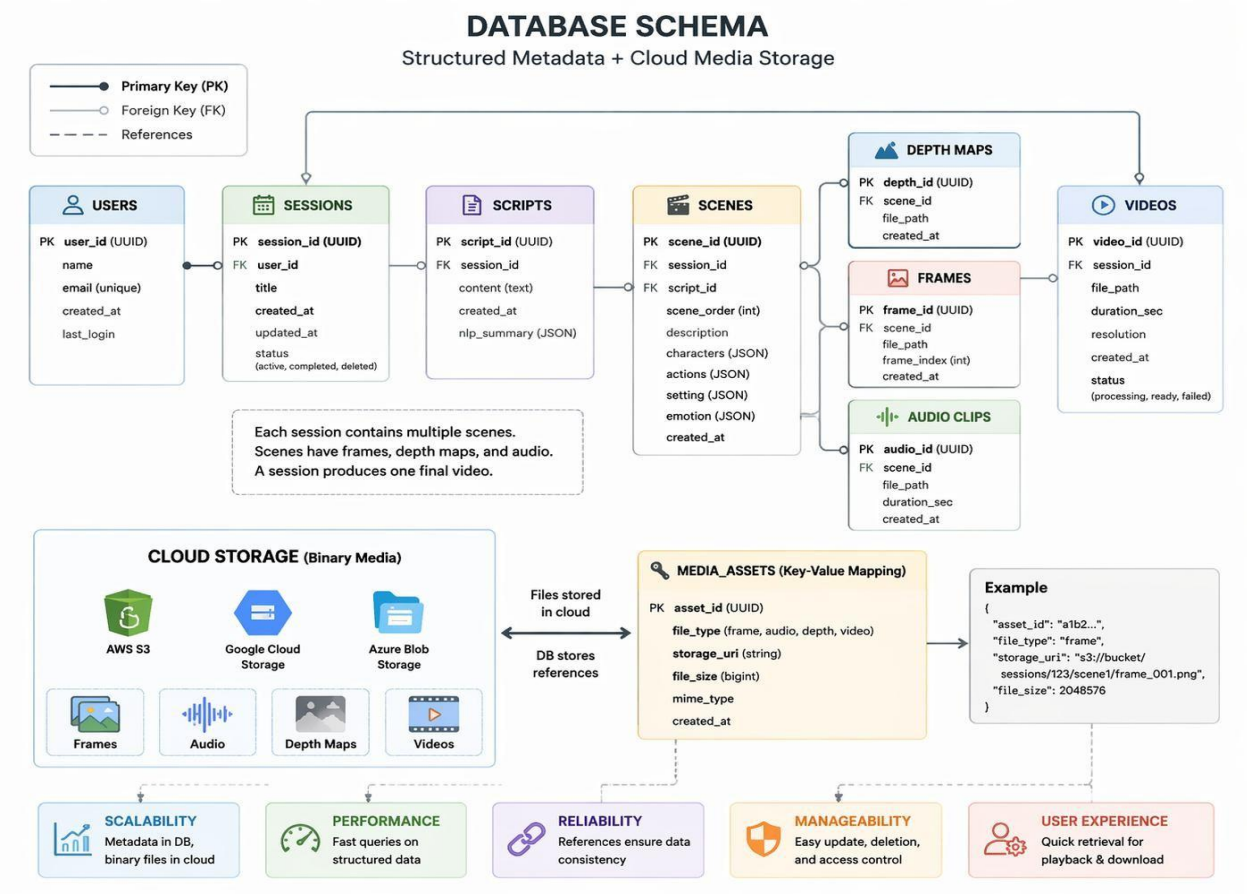

Figure 4: Database Design

V. IMPLEMENTATION

A. Software Requirements

The system requires Windows 10 as the operating system, providing a stable environment compatible with Python, JavaScript, and the required AI and web development frameworks. Python serves as the core backend language for AI model execution and data processing, while JavaScript with React.js powers the interactive frontend web application interface.

Key frameworks include Flask and FastAPI for backend API development, and Node.js with Express.js for server-side routing and middleware management. Essential libraries include NLP transformer models for script analysis, latent diffusion libraries for video generation, and FFmpeg for media processing and video composition.

B. Hardware Requirements

The system requires a modern multi-core processor such as an Intel Core i5 or higher to handle intensive real-time computations associated with AI model inference and video rendering. A dedicated GPU, preferably an NVIDIA RTX 30- series or equivalent, is essential for accelerating deep learning model execution and parallel frame generation tasks.

A minimum of 16 GB RAM is required for loading large AI models and managing intermediate data buffers, with 32 GB recommended for optimal performance during full-scale video generation. At least 500 GB of SSD storage ensures fast data access for model files, generated assets, and intermediate processing outputs throughout the video generation pipeline.

C. Dataset Description

The system is trained and evaluated using large- scale text-video paired datasets that contain diverse textual descriptions aligned with corresponding video sequences across multiple domains. Depth information is derived from standard monocular depth estimation datasets and synthetically generated depth maps using pre-trained depth estimation models.

Training datasets incorporate multi-scene video samples spanning indoor environments, outdoor landscapes, human interactions, and object animations to ensure broad generalization across script themes. Preprocessing pipelines normalize depth values, tokenize text inputs, and align temporal video frames to ensure consistency across all training samples.

D. Preprocessing Implementation

Script preprocessing involves tokenization, sentence segmentation, and keyword extraction using transformer-based NLP models to identify scene boundaries, characters, and action descriptors from raw text input. Depth map preprocessing normalizes raw depth values to a standardized range and resizes spatial maps to match the input resolution requirements of the diffusion model.

Video frame preprocessing includes frame-level normalization, temporal alignment, and augmentation techniques applied to training data to improve model robustness and generalization. The preprocessing pipeline is fully automated and integrated into the backend workflow, ensuring consistent data formatting before passing inputs to the AI generation modules.

E. Model Training

The video generation model is based on a pre- trained latent diffusion architecture extended with temporal attention modules that model inter-frame dependencies to produce temporally coherent video sequences. Depth conditioning is integrated through cross-attention mechanisms that inject depth map embeddings into the diffusion process, guiding spatial layout generation at each denoising step.

Training employs a combination of reconstruction loss and temporal consistency loss functions, optimized using the Adam W optimizer with a cosine learning rate scheduler over multiple training epochs on GPU clusters. Fine-tuning is performed on domain-specific datasets to adapt the model to diverse script themes and improve depth- guided spatial accuracy in the generated outputs.

F. Real-Time Detection

The system incorporates real-time scene segmentation and transition detection during script processing to dynamically adjust video generation parameters for each identified scene segment. A scene changes detection algorithm monitors depth map variations and textual cues to trigger re- conditioning of the diffusion model with updated inputs for each new scene.

Inference optimization techniques including model quantization and batch processing are applied to reduce per-frame generation latency, enabling near real-time performance on high-end GPU hardware. The backend scheduler coordinates parallel processing of independent scenes to further accelerate end-to-end video generation while maintaining temporal consistency across scene boundaries.

G. Alert System

An integrated alert system monitors the video generation pipeline for errors, processing timeouts, and resource bottlenecks, notifying users through the web interface when issues arise. Automated validation checks are applied after each generation stage to detect frame quality degradation, audio synchronization errors, or depth inconsistencies before proceeding to the next step.

The alert system logs all processing events and error conditions to a centralized monitoring dashboard, enabling developers to diagnose issues and optimize pipeline performance over time. Threshold-based alerts are configured for GPU memory usage, processing time limits, and output file size to prevent system failures during large- scale video generation tasks.

H. Dashboard

The web-based dashboard provides users with a centralized interface for script submission, real-time generation progress tracking, video preview, and download management, designed using React.js for responsive interaction. Administrative users can access system-level monitoring features including resource utilization metrics, generation queue status, and error log visualization through dedicated dashboard panels.

The dashboard integrates interactive controls for

adjusting video generation parameters such as depth strength, voice style, and scene transition timing, enabling users to customize outputs without technical expertise. A project management interface allows users to save, revisit, and regenerate previously submitted scripts, supporting iterative refinement of generated video content.

VI. RESULTS AND DISCUSSION

A. Testing Strategy

The system was evaluated using a multi-phase testing strategy encompassing unit testing of individual modules, integration testing of the end- to-end pipeline, and user acceptance testing with non-technical participants. Test scripts spanning diverse domains including nature scenes, indoor environments, human dialogue, and action sequences were used to assess generalization performance.

Ablation studies were conducted to isolate the contribution of depth conditioning by comparing outputs generated with and without depth guidance under identical text input conditions. Performance benchmarks were measured on standardized hardware configurations to ensure reproducibility and fair comparison with baseline and competing methods.

B. Performance Metrics

The system performance was evaluated using Fréchet Inception Distance (FID) to measure visual quality and frame realism, alongside Fréchet Video Distance (FVD) to assess temporal consistency across generated video sequences. Depth accuracy was measured using standard depth estimation metrics including absolute relative error and threshold accuracy scores against ground truth depth maps.

Text-video alignment was assessed using CLIP similarity scores that measure the semantic correspondence between input text prompts and generated video frames. Processing time per video segment and GPU memory consumption were also recorded to evaluate system efficiency under varying input conditions and hardware configurations.

C. Experimental Results

Experimental results demonstrate that the proposed depth-guided video generation system consistently produces spatially accurate and visually coherent outputs across diverse script types compared to text-only baseline models. Depth conditioning significantly reduced spatial inconsistencies, with object placement accuracy improving by a measurable margin over models trained without depth supervision. Temporal consistency metrics showed substantial improvement over frame-independent generation baselines, confirming that the temporal attention modules effectively model inter-frame dependencies during the diffusion process. The integrated TTS voiceover module achieved high audio-video synchronization accuracy across all tested script segments with diverse emotional tones.

D. Analysis

The analysis confirms that integrating depth maps as conditioning signals provides meaningful spatial guidance to the diffusion model, resulting in more realistic 3D scene representations than text-only conditioning approaches. Temporal attention modules prove essential for maintaining object stability and motion continuity across video frames, particularly in scenes with multiple moving elements.

While the system performs well across most script types, complex multi-object scenes with rapid camera movements exhibit minor depth estimation errors that propagate into slight spatial distortions in generated video frames. These limitations are primarily attributed to inaccuracies in monocular depth estimation for occluded objects and fast- moving scene elements.

E. Comparison

Compared to Make-A-Video, the proposed system achieves superior spatial accuracy by explicitly conditioning on depth maps, enabling more precise control over object positioning and 3D scene layout. Against Synthesia and VEED.IO, the proposed system offers significantly greater generative flexibility and depth-aware spatial control, though at higher computational cost.

In comparison to Video Fusion's decomposed diffusion approach, the proposed method demonstrates competitive temporal consistency while adding the advantage of explicit depth-guided spatial conditioning not present in the baseline architecture. The system's modular design also offers greater extensibility for integrating new conditioning signals such as optical flow or semantic segmentation maps in future work.

VII. CONCLUSION AND FUTURE WORK

A. Conclusion

This paper presented a text and depth guided controllable video generation system that integrates NLP-based script analysis with depth-conditioned latent diffusion models to produce spatially accurate and temporally consistent videos from textual inputs. The proposed modular architecture successfully combines scene understanding, depth estimation, AI video generation, speech synthesis, and media composition into a seamless end-to-end pipeline.

Experimental results confirm that depth conditioning significantly improves spatial accuracy and visual realism compared to text-only video generation baselines, validating the effectiveness of the proposed approach. The system demonstrates broad applicability across domains including film production, gaming, virtual reality, digital media, and educational content creation.

B. Future Work

Future work will focus on extending the system to support real-time video generation on consumer-grade hardware through model compression techniques including knowledge distillation, pruning, and quantization. Incorporating optical flow-based motion conditioning and semantic segmentation maps alongside depth information is planned to further enhance scene dynamics and object motion control.

The system will be extended to support multi- modal inputs including reference images and audio prompts, enabling users to guide video generation with richer and more diverse conditioning signals. Additionally, exploring transformer-based video generation architectures as alternatives to latent diffusion models may offer improved computational efficiency and scalability for longer video sequences.

REFERENCES

[1] U. Singer *et al.*, "Make-A-Video: Text-to-video generation without text-video data," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2023.

[2] Z. Luo *et al.*, "VideoFusion: Decomposed diffusion models for high-quality video generation," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 10209–10218.

[3] Kawar *et al.*, "Imagic: Text-based real image editing with diffusion models," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 6007–6017.

[4] M. Zhao, R. Wang, F. Bao, C. Li, and J. Zhu, "ControlVideo: Adding conditional control for one-shot text-to-video editing," arXiv:2305.17098, 2023.

[5] S. Ge *et al.*, "Long video generation with time-agnostic VQGAN and time-sensitive transformer," in *Proc. Eur. Conf. Computer Vision (ECCV)*, 2022, pp. 102–118.

[6] Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[7] P. Kingma and M. Welling, "Auto-encoding variational Bayes," arXiv:1312.6114, 2013.

[8] M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[9] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.

[10] M. H. Sqalli, F. Al-Haidari, and K. Salah, "EDoS-shield: A two-step mitigation technique against EDoS attacks in cloud computing," in *Proc. IEEE/ACM Int. Conf. Utility and Cloud Computing*, 2011.

[11] Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv:1301.3781, 2013.

[13] L. Spitzner, *Honeypots: Tracking Hackers*. Boston, MA, USA: Addison-Wesley, 2003.

[14] X. Shu, D. Yao, and B. G. Ryder, "Privacy-preserving and trustworthy cyberphysical systems," *IEEE Trans. Dependable and Secure Computing*, 2015.

[15] Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. ACM-SIAM Symp. Discrete Algorithms (SODA)*, 2007.