Research Article

VectorDiff: A Manifesto for a Differential, Semantically Rich Vector Animation Format for Scientific and AI-Driven Visualization

Sławomir Wiśniewski[1]

1. SCI4BIZ, Poland

The proliferation of dynamic, multidimensional data in computational science, ranging from medical imaging and molecular simulations to artificial intelligence-generated content, has exposed critical limitations of existing visualization and animation formats. Raster video formats lack scalability and semantic depth, while traditional vector animation standards are often long-winded and unsuited to capturing the subtle, incremental changes inherent in scientific processes. We introduce VectorDiff, a novel differential vector animation format designed to address these challenges. VectorDiff utilizes a declarative JSON-based structure that defines a baseScene and a timeline with explicit, time-stamped transformations. By storing only the difference (delta) between states, it achieves unparalleled performance and data precision. This paper introduces the basic principles of the VectorDiff format, its architecture, and its transformation potential in four key areas: diagnostic medical imaging, molecular dynamics, robotic surgery, and AI-generated content. We argue that VectorDiff is not just an alternative format but a necessary paradigm shift toward a more efficient, precise, and semantically aware representation of dynamic scientific phenomena.

Corresponding author: Slawomir A. Wisniewski, sa.wisniewski@sci4biz.edu.pl

1. Introduction: The crisis of representation in dynamic data

Scientific discovery is increasingly driven by our ability to model and understand dynamic systems. We simulate the coiling of proteins on a picosecond scale, track tumor growth over months based on sequential MRI scans and record every subtle movement of surgical robot tools. In parallel, generative artificial intelligence models are now capable of creating complex, evolving visual narratives[1]. However, the formats we use to store, transmit, and analyze these dynamic processes remain rooted in outdated paradigms.

Raster video formats (e.g., MP4, AVI) are fundamentally inadequate. They represent data as a grid of pixels, discarding the underlying geometric and semantic information. This results in large file sizes, loss of precision when scaling, and an inability to interact with or query individual objects in the scene. On the other hand, declarative vector animation formats, such as SVG with SMIL[2], while maintaining geometric integrity, are often long-winded and burdened by overhead associated with the DOM model. They were designed for illustrative web animation, not for efficient representation of thousands of incremental state changes generated by scientific simulation, leading to significant performance issues with large numbers of dynamic objects.

This gap creates a "crisis of representation." We require a format that is:

- **Differential and efficient**: Capable of storing huge time sequences by encoding only the changes (deltas) between states.
- **Vectorized and precise**: Infinitely scalable and capable of representing complex geometries with mathematical precision.
- **Semantically rich**: Able to maintain the identity and attributes of individual objects throughout the animation, allowing for interaction and analysis.
- **Platform-independent:** Easy to parse and render in both 2D (web) and 3D (VR/AR, scientific software) environments.

This article introduces VectorDiff as a comprehensive solution to this crisis, summarizing its key advantages in the table below.

2. The VectorDiff format: a differential philosophy

The basic philosophy of VectorDiff is to treat animation not as a sequence of discrete, independent frames but as an initial state followed by a precise, time-ordered log of transformations. This is implemented using a simple but powerful JSON structure.

2.1. Major components

The VectorDiff file consists of two primary keys: baseScene and timeline.

baseScene: An array of VectorObject elements that defines the initial state of the scene at time t=0. Each VectorObject has a unique id, type (e.g., rect, ellipse, path), and structured geometric data (e.g. {"x": 10, "y": 20, "width": 50, "height": 50}) and a set of attributes for styling (e.g., fill, stroke).

timeline: an object in which the keys represent discrete timestamps (e.g., "1.05s"), and the values are arrays of transformations that occur at that exact moment. Each transformation targets a specific object by its targetId and describes the change (e.g., translate, rotate, scale).

```
{
"version": "1.0",
"baseScene": {
"objects": [
{
"id": "protein_alpha_helix_1",
"type": "path",
"data": { "d": "M10 80 C 40 10, 65 10, 95 80" },
"attributes": { "fill": "none", "stroke": "#3498db" }
}
]
},
"timeline": {
"0.5": [
{
"targetId": "protein_alpha_helix_1",
"transformation": { "type": "scale", "sx": 1.1, "sy": 1.1 }
}
],
"1.2": [
{
"targetId": "protein_alpha_helix_1",
"transformation": { "type": "rotate", "angle": 15, "centerX": 50, "centerY": 45 }
}
]
}
}
```

2.2. Unified transformation model

A critical design principle in VectorDiff is a unified approach to transformations. All geometric changes applied to an object over time - translation, rotation, scaling, or affine matrices - are cumulatively applied to a single transform attribute in the object state. The underlying geometric data remains constant. This provides a single source of truth about the object's state at any point in time, eliminating ambiguity and ensuring predictable, compositional transforms —a principle directly inherited from standards such as SVG and modern graphics pipelines.

3. System architecture and applications

The VectorDiff ecosystem is designed as a modular monorepository consisting of a core engine, visualization renderers, and high-level application packages.

3.1. Core engine and visualization

@vectordiff/core: A lightweight, dependency-free TypeScript library responsible for parsing, validating, and applying transformations to VectorObject structures. It is a canonical implementation of format logic.

@vectordiff/visualization: Provides renderers that bring the VectorDiff animation to life:

- **SVGRenderer**: A React-based component that renders animation as a standard, interactive SVG, ideal for web dashboards and publishing.
- **ThreeRenderer**: A powerful renderer that interprets VectorDiff data and creates a corresponding 3D scene using Three.js[3]. This enables high-quality, interactive 3D visualization for advanced applications.

3.2. Application I: Medical imaging and diagnostics

In radiology, clinicians analyze sequences of CT or MRI scans to assess disease progression. VectorDiff represents a paradigm shift in the way this data is visualized[4]. Instead of a stack of static images or scraped video, a radiologist can interact with a VectorDiff animation in which segmented structure (e.g., tumors, lesions, blood vessels) is represented as VectorObject. The timeline accurately documents changes in their size, shape, and position during treatment.

This is especially important in the context of modern analytical methods such as LILAC [5], which automatically highlights significant temporal changes between pairs of images. VectorDiff provides an ideal data structure for storing and querying precisely those "temporal change predictions" that such models generate. Moreover, with the clinical implementation of 4D imaging modalities that capture dynamic physiological processes in real-time, such as 4D Flow MRI for cardiac evaluation[6], the need for a format that can effectively encode time-varying geometry, rather than just a series of static 3D volumes, becomes critical.

This approach aligns with the increasing emphasis on longitudinal data analysis in clinical trials, as reflected in research initiatives supported by institutions such as the National Institutes of Health[7].

3.3. Application II: Molecular dynamics and drug design

As molecular dynamics (MD) simulations continue to be a cornerstone of modern drug design[8], the challenge of analyzing the resulting multidimensional trajectory data remains. While early molecular dynamics simulation solutions, such as Anton's specialized machines[9], produced massive amounts of trajectory data, current tools often generate thousands of PDB files or render to non-interactive movies.

VectorDiff, powered by ThreeRenderer, can represent each secondary structure or protein domain as a separate VectorObject. The timeline can then animate conformational changes, ligand binding events, or allosteric effects with high precision and efficiency. This offers a significantly richer analytical experience than simple video playback, enabling new paradigms for interactive exploration and hypothesis testing in virtual reality environments[10].

Researchers can pause the animation at any time, query the exact transformation of a specific domain, and interact with the 3D model. VectorDiff is architecturally poised to support a future in which analysis moves from post-hoc trajectory files to interactive, real-time visualization of conformational and microenvironmental changes[11].

3.4. Application III: Robotic surgery and procedure analysis

In robotic-assisted surgery (e.g., using the DaVinci system), recording and reviewing procedures are essential for training, quality assessment, and providing a forensic opinion. This is an area of growing importance, with a significant increase in AI-based tool tracking analysis as of 2018[12][13].

VectorDiff is the ideal format for this task. The path of each surgical tool can be registered as a VectorObject, and the timeline captures its every movement and rotation with millisecond precision, a key requirement for 3D tool movement metrics that correlate with surgical proficiency[14]. Moreover, safety-critical information such as proximity warnings or virtual forbidden zones can be encoded as other vector objects, creating a complete, reproducible, and analyzable record of the entire surgical procedure.

Such a record provides the basis for surgical digital twins (digital twins) and photorealistic simulations based on real-world data[15].

3.5. Application IV: Canonical format for AI-generated content

As generative models become more proficient at producing visual content, the ideal representation of the output has become a key research question. While early models produced raster images, a new wave of research is exploring the direct synthesis of structured, editable vector graphics to overcome the limitations of pixel-based formats.

Recent work has demonstrated that the direct optimization of traditional vector primitives, such as Bézier curves, is complex and often leads to redundant or incorrect geometries. To address this, methods such as NIVeL[16] propose using an indirect representation based on neural implicit fields that can capture complex topologies while remaining resolution independent.

VectorDiff offers a complementary, declarative approach to the same problem, providing a normalized, easily parsable JSON structure as an alternative to a purely neural representation. Moreover, the challenges of maintaining style consistency have led to the development of advanced multistage pipelines[17] that distill knowledge from image diffusion models to ensure structural regularity in the final vector output.

These efforts underscore the industry's desire to create structured, semantic, and editable generative content. VectorDiff is, therefore, a proposal that is both non-speculative and timely. It provides a canonical, web-native format that aligns with the goals of this cutting-edge research, offering a potential universal target format for generative models that prioritizes interoperability and edibility.

4. Discussion and future directions

VectorDiff presents a distinct set of advantages over existing formats for dynamic scientific data:

- **Efficiency**: File sizes are dramatically reduced for long, incremental animations compared to storing full-state snapshots.
- **Precision and scalability**: As a vector format, it maintains quality at any resolution.
- **Interactivity and semantics**: While preserving the identity of objects, it allows direct interaction, querying, and analysis of individual components inside the animation.

The main challenge for VectorDiff lies in its adaptation. This requires adapting tool chains and scientific software to export and import the new format. Our work provides basic libraries to facilitate this process.

Future work will focus on three key areas:

Hardware acceleration: development of WebGL and GPU-accelerated renderers capable of handling scenes with millions of objects in real-time. The development of GPU-accelerated renderers is critical because standard DOM-based SVG rendering exhibits significant performance degradation when animating thousands of individual objects, a common scenario in scientific visualization.

Standardization: the main challenge for VectorDiff lies in adoption, an obstacle that can be overcome by following the successful model of other open formats. The glTF format, often referred to as "3D JPEG," has achieved wide popularity due to its focus on being a lightweight, interoperable transmission format managed by an industry consortium. We propose a similar path for VectorDiff, positioning it as "glTF for dynamic data." A key first step will be the formation of a W3C Community Group to formalize the specification. This is a low-threshold entry process, requiring only a proposal and the support of five people to initiate a public forum for developing the specification and building consensus.

Direct integration: In parallel with standardization efforts, the creation of plug-ins for major scientific software platforms (e.g., PyMOL[18], ParaView[19], 3D Slicer[20]) to support native VectorDiff import/export is essential to lower the barrier to adaptation.

5. Conclusions

The era of static diagrams and rasterized videos as the primary means of scientific communication is drawing to a close. The complexity and dynamism of modern data require a new representation language. VectorDiff is our proposed manifesto for this new language. It is a format built on principles of efficiency, precision, and semantic richness, designed not only for viewing but also for interacting with and understanding the dynamic processes that define the frontiers of science and technology.

Notes

Github repository: https://github.com/SlawomirWisniewski73/VectorDiff

References

1. ^Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, ... Bengio Y (2014). "Generative adversarial networks." arXiv. doi:10.48550/arXiv.1406.2661.
2. ^W3C (2011). "Scalable Vector Graphics (SVG) 1.1 (Second Edition)." W3C. https://www.w3.org/TR/SVG11/.
3. ^Cabello R, Three.js Authors (2010). "Three.js - JavaScript 3D Library." Three.js. https://threejs.org/.
4. ^Ellingson BM, Bendszus M, Sorensen AG, Pope WB (2014). "Emerging techniques and technologies in brain tumor imaging." Neuro Oncol. **16**(Suppl 7):vii12-23. doi:10.1093/neuonc/nou221.
5. ^Kim H, Karaman BK, Zhao Q, Wang AQ, Sabuncu MR (2025). "Learning-based inference of longitudinal image changes: Applications in embryo development, wound healing, and aging brain." Proc Natl Acad Sci U S A. **122**(8):e2411492122.
6. ^Bria A (2025). "The Evolution of Medical Imaging in 2025: Emerging Trends and Technological Advancements." Medicai Blog. https://blog.medicai.io/en/future-of-medical-imaging/.
7. ^National Institutes of Health (2024). "A Longitudinal Analysis Stream for FreeSurfer." NIH RePORTER.
8. ^Grewal S, Deswal G, Grewal AS, Guarve K (2025). "Molecular dynamics simulations: Insights into protein and protein ligand interactions." Adv Pharmacol. **103**:139-162. doi:10.1016/bs.apha.2025.01.007.
9. ^Shaw DE, Deneroff MM, Dror RO, Kuskin JS, Larson RH, Salmon JK, ... Wazzan N (2008). "Anton, a special-purpose machine for molecular dynamics simulation." Communications of the ACM. **51**(7):91-97.
10. ^Wiśniewski S (2024). "Integration of LigandDesigner systems: ligand design for laboratory applications and HPC environments." Zenodo. doi:10.5281/zenodo.14512310.
11. ^Yang S, Jin S, Zhang M, Chen Y, Guo Y, Hu Y, Wolynes PG, Xiao H (2024). "Real-Time Visualization of Protein Microenvironment Changes with High Spatial Resolution in Live Cells via Site-Specific Incorporation of Rotor-Based Fluorescent Noncanonical Amino Acids." bioRxiv. doi:10.1101/2024.10.19.619218.
12. ^Moglia A, Georgiou K, Georgiou E, Satava RM, Cuschieri A (2021). "A systematic review on artificial intelligence in robot-assisted surgery." Int J Surg. **95**:106151. doi:10.1016/j.ijsu.2021.106151.
13. ^Yangi K, On TJ, Xu Y, Gholami AS, Hong J, Reed AG, Puppalla P, Chen J, Tangsrivimol JA, Li B, Santello M, Lawton MT, Preul MC (2025). "Artificial intelligence integration in surgery through hand and instrument tracking: a systematic literature review." Front Surg. **12**:1528362. doi:10.3389/fsurg.2025.1528362.
14. ^Narasimhan S, Turkcan MK, Ballo M, Choksi S, Filicori F, Kostic Z (2025). "Monocular 3D Tooltip Tracking in Robotic Surgery—Building a Multi-Stage Pipeline." Electronics. **14**(10):2075. doi:10.3390/electronics14102075.

15. ^ *Han JJ, Wu JY (2025). "EndoPBR: Material and Lighting Estimation for Photorealistic Surgical Simulations via Physically-based Rendering." arXiv. doi:10.48550/arXiv.2502.20669.*

16. ^ *Thamizharasan V, Liu D, Fisher M, Zhao N, Kalogerakis E, Lukáč M (2024). "NIVeL: Neural Implicit Vector Layers for Text-to-Vector Generation." 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4589-4597. doi:10.1109/CVPR52733.2024.00439.*

17. ^ *Zhang P, Zhao N, Liao J (2025). "Style Customization of Text-to-Vector Generation with Image Diffusion Priors." ACM Transactions on Graphics (Proceedings of SIGGRAPH 2025). doi:10.48550/arXiv.2505.10558.*

18. ^ *DeLano WL (2002). The PyMOL molecular graphics system, version 1.2r3pre. Schrödinger, LLC.*

19. ^ *Ahrens J, Geveci B, Law C (2005). "ParaView: An end-user tool for large data visualization." The Visualization Handbook. Elsevier. pp. 717-731.*

20. ^ *Fedorov A, Beichel R, Kalpathy-Cramer J, Finet J, Fillion-Robin JC, Pujol S, ... Kikinis R (2012). "3D Slicer as an image computing platform for the Quantitative Imaging Network." Magnetic Resonance Imaging.* ***30****(9):1323-1341.*

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.