Computing the Quality of Isosurface-Based Visualizations Using a Feed-Forward Model for Single-Image Reconstruction

Li-Ta Lo*
Los Alamos National Laboratory, Los Alamos, NM, USA

Christian Heine†
Leipzig University, Leipzig, Germany

Roxana Bujack‡
Los Alamos National Laboratory, Los Alamos, NM, USA

Abstract

Automatic evaluation of visualization quality enables optimization of visualization parameters without human intervention. In isosurface visualization, prior work estimates quality by reconstructing a 3D surface from multiple images using NeRF, followed by volumetric reconstruction and comparison to the original data. In this paper, we replace the NeRF-based multi-view reconstruction with a diffusion-based single-image 3D reconstruction model. This formulation removes the need for auxiliary views and aligns the evaluation setting with single-image visualization scenarios. Because the reconstructed geometry is not produced in the original coordinate system, we introduce an explicit alignment step to resolve global pose and scale ambiguities. Experimental results on benchmark datasets show that the proposed approach reduces surface reconstruction error and improves runtime relative to the NeRF-based pipeline. These results suggest that strong learned priors provide a more reliable reconstruction backend for reconstruction-based evaluation while enabling more efficient workflows.

Index Terms: Scientific visualization, machine learning, isosurface.

1 Introduction

Automatic design of visualizations dates back to at least Mackinlay's Automatic Presentation Tool [27]. Recent progress in AI-driven visualization generation has made it possible to synthesize visualizations directly from natural-language prompts [28, 29]. Most of these systems select a design from a large space of possible alternatives, most of which are suboptimal. Often, the best design not only depends on the type of data to be visualized, but also on the actual data itself. For example, when visualizing volume data in a static image, the goal often is to choose a descriptive subset of the data and a viewing position minimizing occlusions [31]. A generally accepted assessment of a good design is that it preserves enough information to support a truthful mental model of the underlying data [42, 39, 13, 25]. Heine proposed building computational models of visualization processes, comprising both machine and human parts, to quantify visualization properties, such as their quality [17]. Recently, Bujack et al. proposed using machine-learning methods to estimate how accurately humans can reconstruct the underlying data from a visualization [11]. They gave examples of estimating the reconstruction accuracy of different colormaps in 2D and on 3D surfaces, as well as volumetric scalar fields from images showing a single isosurface color-coded with its isovalue. Their implementation used NeRF [30] to recover a 3D surface mesh from a 3×3 matrix of images, each of which depicts a single isosurface, but from viewing angles of $\pm 10^\circ$ in spherical coordinates around the original viewing angle. This augmentation was not desired but necessary because NeRF requires multiple images to have sufficient depth cues to reconstruct the surface. Without the auxiliary views, it often did not converge or produced implausible results. The recovered surface then had to be further processed to reconstruct the original volumetric scalar field from which the isosurface originated, and using the L_2 norm, judge how well this reconstructed field resembles the original data. This score is used in a larger system that tests combinations of isovalues, colormaps, and viewing angles and selects the best combination.

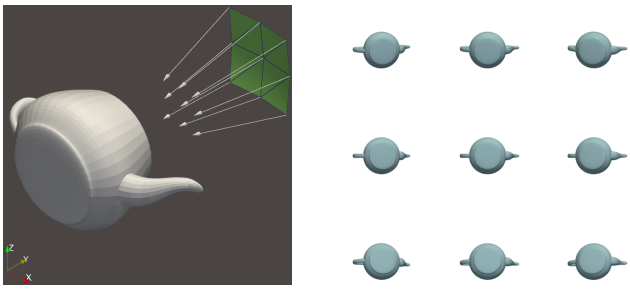


Figure 1: Bujack et al. sought to reconstruct data from only the center image but needed to use nine images because NeRF would not produce reliable results from a single image.

In this paper, we focus on the subproblem of reconstructing 3D surfaces from 2D images and address one specific shortcoming of the previous workflow, namely NeRF's reliance on multiple input images. Recent image-to-3D reconstruction models such as Microsoft's Trellis [43] learn strong geometric priors from large-scale 3D datasets and can reconstruct plausible geometry from a single image. This capability is particularly attractive for reconstruction-based visualization evaluation because it removes the need for auxiliary viewpoints required by NeRF while matching the information available in many static visualization scenarios. While NeRF requires camera angles to accompany the input images and can reconstruct geometry in the original coordinate system, Trellis does not need or make use of such information and its output has its own coordinate system. Since the downstream geometric comparison assumes that reconstructed and reference meshes share a common coordinate frame, the generated mesh cannot be compared directly with the ground-truth geometry without first resolving the global pose and scale ambiguity. We address this problem and present a method to align the coordinate systems of the input isosurface and its reconstruction. Despite this additional step, the resulting system is both faster and more accurate in terms of surface reconstruction. The contributions of this paper are

- a Trellis-based reformulation of the reconstruction metric for single-image 3D isosurface evaluation that uses only one instead of the previous nine images,

*e-mail:ollie@lanl.gov
†e-mail:heine@informatik.uni-leipzig.de
‡e-mail:bujack@lanl.gov

- an explicit alignment step for resolving global pose/scale ambiguities in generated meshes prior to error computation, and
- experiments demonstrating improvements over the approach of Bujack et al. with respect to accuracy and run time.

2 Related Work

AI-generated visualization. Recent advances in large language models have enabled visualization systems that synthesize visualizations directly from natural-language input. Representative examples include language-to-visualization systems such as Chat2VIS [28], ChatVis [29, 33], and ncNet [26]; mixed-initiative visual analytic systems such as LightVA [44] and PhenoFlow [23]; multi-agent visualization pipelines such as VisPath [34] and PlotGen [16]; and scientific visualization systems including VizGenie [6] and NLI4VolVis [1]. These systems automate visualization generation but generally rely on users to assess the quality of generated results. Our work instead focuses on automatic evaluation, enabling reconstruction-based quality assessment as a component of AI-assisted visualization workflow.

Visualization quality metrics. Visualization quality has been studied through perceptual and information-theoretic heuristics [4, 3, 7]. These methods are often effective but typically tied to specific data types, tasks, or encoding assumptions. The idea that a good visualization should preserve recoverable information has strong precedent. Mackinlay analyzed expressiveness and effectiveness in relation to recoverable data [27]. Jänicke et al. introduced visual reconstructability as an explicit metric for 2D flow visualization [21]. Related information-channel perspectives were developed by Bramon et al. [9] and Chen et al. [14]. Bujack et al. extended this idea to arbitrary visualization metaphors and 3D via machine-learned reconstruction [11].

Isocontour and viewpoint selection. Meaningful isovalue selection has long been central to 3D scalar visualization quality, but most approaches remain heuristic, such as placing contours near boundaries [24], emphasizing topological features derived from contour trees [37, 12], or maximizing mutual information [10].

Most viewpoint selection techniques also optimize heuristic criteria, e.g., entropy [41, 8], occlusion [31], and feature visibility [38, 36]. However, they disregard interactions between visualization parameters, optimizing parameters directly using predefined objective functions, whereas the reconstructability framework of Bujack et al. evaluates visualizations based on how much information about the underlying data can be recovered. Our work retains this framework while improving its reconstruction backend.

3 Method

Our method consists of three components: replacing the NeRF reconstruction backend in Bujack et al. [11] with the Trellis feed-forward image-to-3D model, aligning the reconstructed geometry with the reference mesh to resolve the global pose and scale ambiguity introduced by single-image reconstruction, and computing the reconstruction signal within the otherwise unchanged framework of Bujack et al. The remainder of this section describes these three components.

3.1 Feed-forward Reconstruction

Feed-forward image-to-3D reconstruction models differ fundamentally from optimization-based methods such as NeRF. Rather than recovering geometry through multi-view optimization, they learn strong geometric priors from large collections of three-dimensional objects and directly infer plausible geometry from a single image. These learned priors enable high-quality reconstruction of objects that resemble those encountered during training while eliminating the need for the auxiliary viewpoints required by NeRF.

In the reconstructability framework of Bujack et al., the reconstruction backend provides the intermediate signal from which visualization quality is subsequently estimated. A more faithful reconstruction backend therefore reduces backend-induced reconstruction error and provides a cleaner reconstruction signal to the otherwise unchanged framework. Our objective is not to redefine the reconstructability metric itself, but to determine whether replacing the NeRF reconstruction backend with a feed-forward model yields a stronger reconstruction signal under the same evaluation framework.

We use the `TrellisImageTo3DPipeline` from Microsoft Trellis with the pre-trained `Trellis-image-large` model (1.2B parameters). Although Trellis supports multiple input images, we use a single input image to match the single-image evaluation setting. For reproducibility, we fix the random number generator seed.

The pipeline produces both a 3D Gaussian splat representation and a mesh reconstruction of the scene. Since the downstream alignment and evaluation pipeline operates on meshes, we use the reconstructed mesh throughout the remainder of our workflow. The mesh is exported as a `GLB` file [20] using Trellis's built-in utility, simplified (ratio = 0.95), and the texture resolution is limited to 1024. We then bake the texture into the vertex properties as RGBA colors using our own custom script rather than Trellis's built-in functionality, and export the results as VTK PolyData for subsequent alignment and evaluation.

3.2 Alignment of Reconstructed Geometry

Single-image 3D outputs from generative models are ambiguous under global rotation, translation, reflection-like axis sign conventions, and isotropic scale [19]. However, direct geometric comparison using metrics such as the symmetric Chamfer distance [2] require consistent coordinate-frames. We therefore apply a deterministic alignment stage prior to scoring.

We explored several direct encoding strategies for alignment by modifying the input images, including rendering directional cues (e.g., colored arrows along coordinate axes) and visualizing dataset bounding boxes. In practice, Trellis frequently omitted such cues or reinterpreted them (e.g., treating bounding box edges as surfaces and merging them with the object). This behavior suggests that Trellis does not reliably preserve explicit geometric annotations and instead tends to simplify or reinterpret structures to produce visually plausible outputs that do not necessarily agree with the input.

Ultimately, we implemented an alignment step. In contrast to Huang et al. [19], who had to rely on 2D-to-3D point correspondences, our task is easier: we can make use of standard alignment algorithms because we know the ground-truth 3D geometry. Given reference points $P \in \mathbb{R}^{N\times 3}$ and reconstructed points $Q \in \mathbb{R}^{M\times 3}$, we first subtract centroids to remove translation. Principal axes are then computed via singular value decomposition (SVD), i.e., $X = U\Sigma V^\top$ [15]. The right singular vectors define orthonormal principal component analysis (PCA) frames for both point sets.

To resolve axis-order and sign ambiguities, we test all 3! axis permutations and 2^3 sign configurations, yielding 48 candidate orientations, i.e., the isometries of a cube. For each hypothesis, we estimate a uniform scale from variance ratios and compute the corresponding full-similarity transformation consisting of rotation/reflection, translation, and scaling.

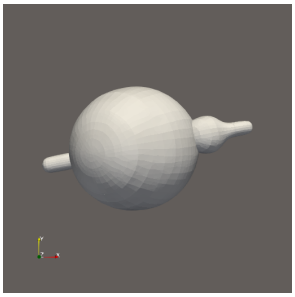
Starting from each initialization (or only the best one), we perform point-to-point iterative closest point (ICP) refinement [5]. At every iteration, nearest neighbors are computed and the optimal rigid (or similarity) transform is obtained in closed form using the SVD-based solution of Kabsch [22] extended with scale following Umeyama [40].

Selecting the alignment that minimizes Chamfer distance while ignoring color and lighting can overfit because an incorrect alignment can accidentally reduce geometric distance for poor recon-

(a) Ground truth at $AZ = 0°$, $EL = 0°$.

(b) Better-performing NeRF reconstruction, $AZ = 0°$, $EL = 0°$.

(c) Typical Trellis reconstruction, $AZ = 0°$, $EL = 0°$.



(d) Trellis failure case, $AZ = 220°$, $EL = -90°$.

Figure 2: Comparison of reconstruction quality of NeRF vs. Trellis through representative examples. Despite using only a single image, Trellis produces significantly better results.

structions. To explicitly include color, lighting, and camera position information, we render each aligned reconstruction from the original camera viewpoint and select the alignment with the highest structural similarity (SSIM) [18] to the input image. This strategy does not eliminate all ambiguities, but substantially improves alignment robustness in practice, particularly for symmetric objects or severely degraded reconstructions.

4 Results

We applied the reconstructability framework of Bujack et al., replacing only the NeRF reconstruction stage with our Trellis-based reconstruction and alignment pipeline while leaving all subsequent stages unchanged. We evaluate the resulting pipeline on the two datasets used by Bujack et al.

4.1 Teapot

The Utah teapot is a well-known 3D model and serves as a benchmark for evaluating surface reconstruction quality. Using PyVista [35], we generated images from multiple viewing angles by combining 19 elevation values (from $-90°$ to $90°$ in $10°$ increments) with 19 azimuth values (from $0°$ to $360°$ in $20°$ increments). Unless otherwise stated, azimuth (AZ) and elevation (EL) are measured relative to the default camera orientation used for rendering (PyVista `camera_position="xy"`). The surface color was set to light blue (hex: #ADD8E6FF). Images were rendered with a transparent background at a resolution of 512×512 pixels.

Figure 2 provides representative visual comparison between the NeRF- and Trellis-based pipelines. Figure 2b shows one of the better-performing NeRF reconstruction, while Figure 2c shows the corresponding Trellis reconstruction. Quantitatively, the mean Chamfer distance across all evaluated viewpoints decreases from 5.54×10^{-3} with the method of Bujack et al. to 5.98×10^{-4} with our approach, corresponding to an 89% reduction ($\sim 9.3\times$ lower surface reconstruction error). Reconstruction error decreases for

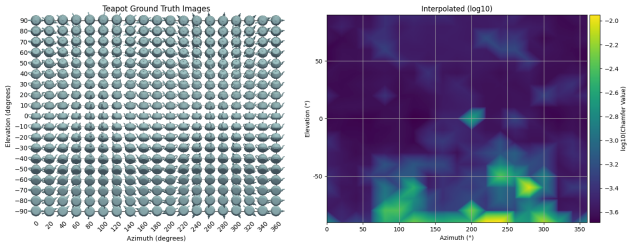


Figure 3: Trellis is able to produce very good reconstruction of the teapot except when it is seen from the bottom. Left: the ground truth renderings of the teapot from different angles. Right: interpolated Chamfer distance to the reconstruction (lower is better).

357 of the 361 (98%) evaluated viewpoints, indicating that the improvement is robust across nearly the entire viewing sphere.

Only four of the 361 evaluated viewpoints exhibit higher reconstruction error than the NeRF-based pipeline. These viewpoints are intrinsically difficult. For example, bottom views provide little information about the teapot handle and its loop topology, making them challenging for both NeRF and Trellis. In Figure 2d, the reconstructed handle no longer forms a closed loop, resulting in a topological reconstruction error that removes one of the object's strongest asymmetric features and makes the reconstructed object approximately axially symmetric. Consequently, the reconstruction provides insufficient geometric constraints for reliable alignment, further increasing the measured Chamfer distance.

Figure 2d also illustrates a limitation of our alignment procedure. When the reconstructed shape is poor or exhibits symmetry, the PCA/ICP-based alignment may fail to uniquely determine the orientation of the non-dominant axes. Multiple rotations around the principal axis can produce geometrically equivalent configurations, introducing ambiguity during alignment. Consequently, the method may converge to an incorrect orientation even when the reconstructed geometry is largely accurate, increasing the measured surface distance. Some of the observed error may therefore result from unresolved rotational ambiguity during alignment rather than substantial reconstruction errors.

We further study how reconstruction quality varies with camera viewpoint. Figure 3 shows the Chamfer distance across the sampled viewing sphere for our method. Reconstruction quality varies more strongly with elevation than with azimuth angle. Views at moderate elevations produce the lowest Chamfer distances, while viewpoints at extreme elevations produce higher reconstruction error. Consistent with observations by Bujack et al., views from below the teapot yield the largest Chamfer distances because of insufficient geometric cues and stronger self-occlusion. Across most azimuth angles, the error distribution remains low, indicating that Trellis's learned priors successfully infer object parts that are not directly visible from the input image. At the same time, the continued degradation for bottom views shows that these priors do not eliminate the dependence of reconstruction quality on viewpoint. Instead, they reduce backend-induced reconstruction error while remaining sensitive to the information available in the visualization.

4.2 Asteroid

We also used the yA31 All Scalars Asteroid Impact dataset, an ensemble of outputs from simulations of an asteroid impacting the ocean [32]. We used timestep 52, representing the state shortly after impact, of the scalar field TEV (temperature in electron volts). Images were generated for the Cartesian product of five isovalues (0.082, 0.209, 0.335, 0.462, 0.588), evenly distributed across the range $[0.019, 0.652]$, four colormaps (cool-warm, spectral, viridis, and rainbow), five elevation angles from $-90°$ to $90°$ in $45°$ incre-

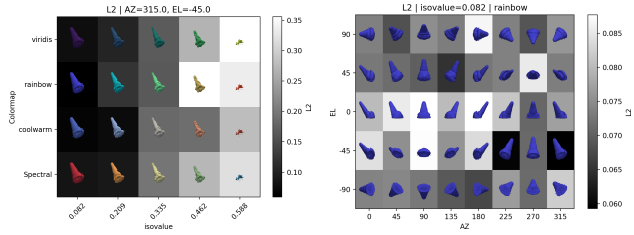
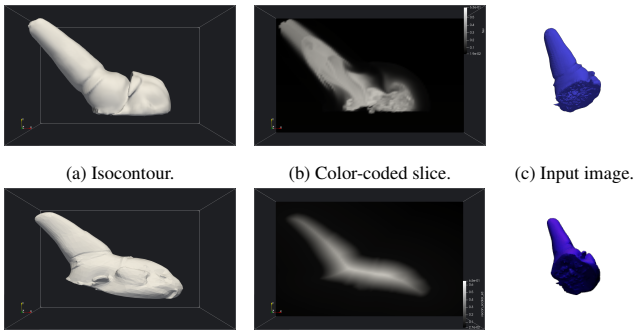


Figure 4: Reconstruction error of our approach shown as 2D slices over pairs of parameters. Grayscale encodes the L_2 distance (darker indicates lower error), ground-truth isocontour images overlaid.



(a) Isocontour. (b) Color-coded slice. (c) Input image.

(d) Isocontour. (e) Color-coded slice. (f) Rec. image.

Figure 5: Top: Original. Bottom: Best reconstruction (rainbow, iso $= 0.082$, AZ $= 315°$, EL $= -45°$).

ments, and eight azimuth angles from $0°$ to $315°$ in $45°$ increments. Isosurfaces and images were generated using PyVista, with surface color determined by the colormapped isovalue. All images were rendered with a transparent background at a resolution of 512×512 pixels.

Figure 4 visualizes reconstruction error as two-dimensional slices over pairs of parameters (isovalue vs. colormap and azimuth vs. elevation), with the remaining two parameters fixed. Lower isovalues yield lower reconstruction error, while higher isovalues increase L_2 distance across most parameter combinations. Quantitatively, the mean L_2 distance decreases from 0.222 for the method of Bujack et al. to 0.192 with our approach, corresponding to an overall improvement of 13.3%. Across colormaps, the mean ranges from 0.186 to 0.199, yielding improvements of approximately 12.0%–15.5%.

Variation across azimuth and elevation is much smoother and more symmetric than those produced by NeRF, suggesting physically more meaningful changes in reconstruction quality with viewpoint. The best performer was AZ $= 315°$ and EL $= -45°$. Differences between colormaps are smaller than variation across isovalues or angles, with *rainbow* achieving the global minimum. The maximum L_2 distance decreases from 0.545 to 0.419, indicating a reduction in worst-case error.

A small fraction of configurations exhibits higher reconstruction error than the NeRF-based pipeline, with the largest regression occurring for the coolwarm colormap at iso $= 0.462$, AZ $= 45°$, and EL $= -90°$. Despite these isolated regressions, the mean and worst-case L_2 distances are both reduced, and reconstruction improves for 75.8% of evaluated configurations.

The best-performing configuration was obtained for the rainbow colormap at iso $= 0.082$, AZ $= 315°$, EL $= -45°$. Figure 5 compares the corresponding input visualization with the reconstructed isosurface from Trellis and the scalar field subsequently recovered from that isosurface, illustrating the level of fidelity achieved by the complete reconstruction pipeline.

Overall, our approach reduces mean and worst-case L_2 distance and improves the majority of configurations, while introducing some regressions in a smaller subset of cases. Similar to the teapot benchmark, the improved reconstruction accuracy on the asteroid dataset indicates that the Trellis-based reconstruction backend also generalizes to a representative scientific visualization while remaining responsive to visualization parameters.

4.3 Runtime

Depending on the complexity of the scene, the combined reconstruction and post-processing times vary. In most cases, the full reconstruction pipeline completes in under one minute on an NVIDIA RTX 4000 Ada-generation GPU with 20 GB of memory. In contrast, NeRF-based reconstruction performs approximately 1,000 optimization iterations per minute, with around 3,000–5,000 iterations required to achieve satisfactory results on the same hardware platform. The time to produce the alignment, in the range of 15 to 30 seconds, depends on the number of iterations the Kabsch algorithm requires to converge. Overall, replacing the NeRF reconstruction backend with Trellis reduces the runtime of the reconstruction stage by approximately a factor of three.

5 Conclusions

This short paper updates the reconstruction-based visualization evaluation framework by Bujack et al. with a Trellis-based single-image 3D reconstruction backend. Experiments on both the Utah teapot and a representative scientific visualization demonstrate that the Trellis-based reconstruction backend improves reconstruction accuracy and runtime while remaining sensitive to viewpoint and visualization parameters.

The key technical addition beyond model replacement is alignment-aware evaluation. Because generated meshes can be accurate but globally misregistered, we explicitly align each reconstruction using our PCA-initialized, ICP-refined Kabsch pipeline before computing reconstruction error. This reduces metric instability and makes geometry-based scoring of isocontour visualizations more reliable.

We still observe occasional misalignment in cases with poor or symmetric reconstructions, as well as cases where alignment may overfit. In future work, we will explore 3D reconstruction approaches that incorporate both strong priors and camera coordinate information during inference.

Our work improves the computational reconstruction stage within the reconstructability framework. Whether learned reconstruction models can be trained or fine-tuned to better predict human reconstructability remains an open question. Such a model could align reconstruction-based evaluation more closely with human interpretation while retaining the computational efficiency of feed-forward inference.

Acknowledgments

We thank Kei Davis for his feedback on this manuscript. This work is published under LANL LA-UR-26-23829. It was supported by the Advanced Simulation and Computing (ASC) Program under grant number DE-SCL0000089. ChatGPT-5.4 and Claude Sonnet 4.5 were used in the writing of this paper, in particular for coding assistance, data processing, and language correction. Everything was human checked.

References

[1] K. Ai, K. Tang, and C. Wang. NLI4VolVis: Natural language interaction for volume visualization via LLM multi-agents and editable 3D Gaussian splatting. *IEEE Transactions on Visualization and Computer Graphics*, 32(1):46–56, 2026. 2

[2] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. Technical Report 153, Artificial Intelligence Center, SRI International, 1977. 2

[3] M. Behrisch, M. Blumenschein, N. W. Kim, L. Shao, M. El-Assady, J. Fuchs, D. Seebacher, A. Diehl, U. Brandes, H. Pfister, T. Schreck, D. Weiskopf, and D. A. Keim. Quality metrics for information visualization. *Computer Graphics Forum*, 37(3):625–662, 2018. 2

[4] E. Bertini, A. Tatu, and D. Keim. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2203–2212, 2011. 2

[5] P. J. Besl and N. D. McKay. Method for registration of 3-D shapes. In *Sensor fusion IV: control paradigms and data structures*, vol. 1611, pp. 586–606. SPIE, 1992. 2

[6] A. Biswas, T. L. Turton, N. R. Ranasinghe, S. Jones, B. Love, W. Jones, A. Hagberg, H.-W. Shen, N. DeBardeleben, and E. Lawrence. VizGenie: Toward self-refining, domain-aware workflows for next-generation scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 32(1):1021–1031, 2026. 2

[7] F. Bolte and S. Bruckner. Measures in visualization space. In *Foundations of Data Visualization*, pp. 39–59. Springer, 2020. 2

[8] U. D. Bordoloi and H.-W. Shen. View selection for volume rendering. In *VIS 05. IEEE Visualization, 2005.*, pp. 487–494, 2005. 2

[9] R. Bramon, M. Ruiz, A. Bardera, I. Boada, M. Feixas, and M. Sbert. An information-theoretic observation channel for volume visualization. *Computer Graphics Forum*, 32(3pt4):411–420, 2013. 2

[10] S. Bruckner and T. Möller. Isosurface similarity maps. *Computer Graphics Forum*, 29(3):773–782, 2010. 2

[11] R. Bujack, L.-T. Lo, E. Stam, A. Biswas, and D. Rogers. The truth, the whole truth, and nothing but the truth: Automatic visualization evaluation from reconstruction quality. In *Computer Graphics Forum*, p. e70447. Wiley, 2026. 1, 2

[12] H. Carr, J. Snoeyink, and M. Van De Panne. Flexible isosurfaces: Simplifying and displaying scalar topology using the contour tree. *Computational Geometry*, 43(1):42–58, 2010. 2

[13] M. Chen, L. Floridi, and R. Borgo. What is visualization really for? *The Philosophy of Information Quality*, pp. 75–93, 2014. 1

[14] M. Chen and A. Golan. What may visualization processes optimize? *IEEE Transactions on Visualization and Computer Graphics*, 22(12):2619–2632, 2015. 2

[15] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. 2

[16] K. Goswami, P. Mathur, R. Rossi, and F. Dernoncourt. PlotGen: Multi-agent LLM-based scientific data visualization via multimodal retrieval feedback. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, pp. 1672—1676. ACM, 2025. 2

[17] C. Heine. Towards modeling visualization processes as dynamic Bayesian networks. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1000–1010, 2021. 1

[18] A. Hore and D. Ziou. Image quality metrics: PSNR vs. SSIM. In *20th International Conference on Pattern Recognition*, pp. 2366–2369. IEEE, 2010. 3

[19] B. Huang, H. Duan, Y. Zhao, Z. Zhao, Y. Ma, and S. Gao. CUPID: Generative 3D reconstruction via joint object and pose modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12741–12752, 2026. 2

[20] ISO/IEC JTC 1. Information technology — runtime 3D asset delivery format — Khronos glTF™ 2.0. Technical Report ISO/IEC 12113:2022, International Organization for Standardization, 2022. 2

[21] H. Jänicke, T. Weidner, D. Chung, R. S. Laramee, P. Townsend, and M. Chen. Visual reconstructability as a quality metric for flow visualization. *Computer Graphics Forum*, 30(3):781–790, 2011. 2

[22] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Foundations of Crystallography*, 32(5):922–923, 1976. 2

[23] J. Kim, S. Lee, H. Jeon, K.-J. Lee, H.-J. Bae, B. Kim, and J. Seo. PhenoFlow: A human-LLM driven visual analytics system for exploring large and complex stroke datasets. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):470–480, 2025. 2

[24] G. Kindlmann and J. W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of the 1998 IEEE Symposium on Volume Visualization*, pp. 79–86, 1998. 2

[25] G. Kindlmann and C. Scheidegger. An algebraic process for visualization design. *IEEE transactions on visualization and computer graphics*, 20(12):2181–2190, 2014. 1

[26] Y. Luo, N. Tang, G. Li, J. Tang, C. Chai, and X. Qin. Natural language to visualization by neural machine translation. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):217–226, 2021. 2

[27] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions On Graphics (TOG)*, 5(2):110–141, 1986. 1, 2

[28] P. Maddigan and T. Susnjak. Chat2VIS: Generating data visualizations via natural language using ChatGPT, Codex and GPT-3 large language models. *IEEE Access*, 11:45181–45193, 2023. 1, 2

[29] T. Mallick, O. Yildiz, D. Lenz, and T. Peterka. ChatVis: Automating scientific visualization with a large language model. In *Proceedings of the SC '24 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis*, SC-W '24, p. 49–55. IEEE Press, 2025. 1, 2

[30] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[31] K. Mühler, M. Neugebauer, C. Tietjen, and B. Preim. Viewpoint selection for intervention planning. In *Proceedings of the 9th Joint Eurographics / IEEE VGTC Conference on Visualization*, pp. 267–274. Eurographics Association, 2007. 1, 2

[32] J. Patchett and G. Gisler. Deep water impact ensemble data set. Technical Report LA-UR-17-21595, Los Alamos National Laboratory, 2017. 3

[33] T. Peterka, T. Mallick, O. Yildiz, D. Lenz, C. Quammen, and B. Geveci. ChatVis: Large language model agent for generating scientific visualizations. In *2025 IEEE 15th Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 22–32, 2025. 2

[34] W. Seo, D. Kang, H. An, T. Kim, S. Cho, S. Lee, M. Yu, J. Park, Y. Bu, and S. Lee. Automated visualization code synthesis via multi-path reasoning and feedback-driven optimization, 2026. arXiv:2502.11140. 2

[35] C. Sullivan and A. Kaszynski. PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK). *Journal of Open Source Software*, 4(37):1450, 2019. 3

[36] S. Takahashi, I. Fujishiro, Y. Takeshima, and T. Nishita. A feature-driven approach to locating optimal viewpoints for volume visualization. In *VIS 05. IEEE Visualization, 2005.*, pp. 495–502, 2005. 2

[37] S. Takahashi, Y. Takeshima, and I. Fujishiro. Topological volume skeletonization and its application to transfer function design. *Graphical Models*, 66(1):24–49, 2004. 2

[38] J. Tao, J. Ma, C. Wang, and C.-K. Shene. A unified approach to streamline selection and viewpoint selection for 3D flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):393–406, 2012. 2

[39] E. R. Tufte and P. R. Graves-Morris. *The Visual Display of Quantitative Information*. Graphics Press, 1983. 1

[40] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 2002. 2

[41] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Viewpoint selection using viewpoint entropy. In *Proceedings of the Vision Modeling and Visualization Conference 2001*, p. 273–280, 2001. 2

[42] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 4th ed., 2019. 1

[43] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang. Structured 3D latents for scalable and versatile 3D generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 21469–21480, 2025. 1

[44] Y. Zhao, J. Wang, L. Xiang, X. Zhang, Z. Guo, C. Turkay, Y. Zhang, and S. Chen. LightVA: Lightweight visual analytics with LLM agent-based task planning and execution. *IEEE Transactions on Visualization and Computer Graphics*, 31(9):6162–6177, 2025. 2