A variational view of visually correctable Gaussian world models

D. Shpylka[1]
[1]Kyiv Academic University, Kyiv, Ukraine
d.shpylka@kau.edu.ua

The purpose of this work is to give a mathematical overview and variational interpretation of a proposed method for physically embodied Gaussian splatting [1]. The original method introduces a dual Gaussian–particle representation, where particles support physical simulation and attached three-dimensional Gaussians provide a differentiable visual representation. This enables a robotic world model that can both predict future states and correct itself using camera observations.

Modern robotic systems require internal world models that are not only visually accurate, but also dynamically meaningful. Classical learned world models often represent the environment in a latent space, while physics-based simulators provide interpretable dynamics but may suffer from model error and imperfect synchronization with real observations. The main idea discussed here is to view visually correctable Gaussian world models as prediction-correction systems: a physical simulator gives a prior forecast of the next state, while differentiable Gaussian rendering defines an observation operator whose residual with respect to the real image produces a correction signal.

Let the state of a Gaussian world model at time t be

$$x_t = \{p_i(t), v_i(t), q_i(t), \Sigma_i(t), c_i(t), \alpha_i(t)\}_{i=1}^{N}.$$

Here $p_i(t)$ and $v_i(t)$ denote particle position and velocity, $q_i(t)$ denotes orientation or other physical attributes, while $\Sigma_i(t)$, $c_i(t)$, and $\alpha_i(t)$ are Gaussian covariance, appearance, and opacity. The physical prediction is

$$x_{t+1}^{\text{pred}} = \Phi_{\Delta t}(x_t, a_t), \tag{1}$$

where a_t is an external action and $\Phi_{\Delta t}$ is a discrete-time physical evolution operator. The visual state is connected to image observations by a differentiable renderer

$$\widehat{I}_t = R(x_t), \tag{2}$$

where R is the Gaussian splatting renderer and I_t is the observed image.

We define visual correction as the following constrained variational problem:

$$x_t^* = \arg\min_{z \in \mathcal{C}} \left[\lambda_{\text{vis}} \ell(R(z), I_t) + \lambda_{\text{phys}} \|z - x_t^{\text{pred}}\|^2 + \lambda_{\text{reg}} \mathcal{R}(z) \right]. \tag{3}$$

Here z is a candidate state, while x_t^* is the corrected state. The first term enforces visual consistency, the second term keeps the corrected state close to the physical forecast, and the third term regularizes the solution. The constraint set $\mathcal{C}$ may encode rigidity, non-penetration, object-level consistency, or other physically meaningful restrictions.

In practice, problem (3) may be solved approximately by a number of projected gradient steps:

$$z^{k+1} = \Pi_{\mathcal{C}} \left(z^k - \eta \nabla_z E(z^k) \right), \tag{4}$$

where E denotes the energy in (3). In particular, the visual component of the update has the form

$$F_{\text{vis}} = -\nabla_z \ell(R(z), I_t). \tag{5}$$

Thus, the visual force used in visually correctable Gaussian world models can be interpreted as the negative gradient of an observation loss.

Remark 1. A visually correctable Gaussian world model can be interpreted as a prediction-correction state-estimation scheme: the physical simulator provides a dynamical prior, the Gaussian renderer acts as a differentiable observation operator, and the image residual defines a correction force on the underlying physical state.

This formulation connects visually correctable Gaussian world models with classical prediction-correction methods. Physical simulation plays the role of a dynamical prior, Gaussian rendering acts as a differentiable observation operator, and image residuals define measurement errors. Thus, visual correction can be understood as an analysis step in which a predicted physical state is adjusted using visual evidence.

The differentiability of the renderer is essential: it allows image-space errors to be propagated back to physical variables. Related principles appear in differentiable simulation and differentiable rendering [4]. The specificity of Gaussian world models is that the observation operator is given by Gaussian splatting, which combines explicit geometric structure with real-time differentiable rendering [2]. Dynamic extensions further show that Gaussian primitives can be treated as persistent moving scene elements [3].

The variational view also suggests efficiency-aware extensions. Since visual correction requires repeated rendering and optimization, memory footprint and rendering speed are essential. Compact Gaussian splatting methods reduce cost by pruning redundant primitives, adapting appearance representation, and quantizing Gaussian attributes [5]. Such ideas may be included in $\mathcal{R}(z)$ as penalties for redundancy, excessive appearance complexity, or unnecessarily high-precision parameters.

Therefore, Gaussian splatting can be viewed not only as a rendering method, but also as a differentiable observation operator for correcting physical world models. The proposed variational viewpoint explains visual correction as gradient-based state estimation and motivates future work on robust losses, adaptive physical-visual weighting, and efficiency-aware regularization for real-time robotic applications.

[1] Abou-Chakra J., Rana K., Dayoub F., Sunderhauf N., Physically Embodied Gaussian Splatting: A Realtime Correctable World Model for Robotics, (2024), arXiv:2406.10788.

[2] Kerbl B., Kopanas G., Leimkuehler T., Drettakis G., 3D Gaussian Splatting for Real-Time Radiance Field Rendering, *ACM Transactions on Graphics* **42** (2023), no. 4, 1–14.

[3] Luiten J., Kopanas G., Leibe B., Ramanan D., Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis, (2023), arXiv:2308.09713.

[4] Jatavallabhula K. M. et al., gradSim: Differentiable Simulation for System Identification and Visuomotor Control, (2021), arXiv:2104.02646.

[5] Papantonakis P., Kopanas G., Kerbl B., Lanvin A., Drettakis G., Reducing the Memory Footprint of 3D Gaussian Splatting, *Proceedings of the ACM on Computer Graphics and Interactive Techniques* **7** (2024), no. 1, 1–17.