RhythmGuassian: Repurposing Generalizable Gaussian Model For Remote Physiological Measurement

Anonymous ICCV submission

Paper ID 4404

1. The Details of Method

The Details of 4D Vitual Camera. The virtual 4D camera is proposed to render face video. The 4D virtual cameras use fixed virtual camera intrinsic parameters E and extrinsic parameters K to render the face video instead of using real camera parameters. The camera intrinsic parameters E use the face center as the focal point use half the size of the face as the focal length, and use an identity matrix for the extrinsic parameters K. The virtual 4D camera initialization is shown in the Tab. 1. Based on such initialization, the virtual 4D camera will predict the 4D Gaussian Map prediction into video.

Algorithm 1 Virtual 4D Camera Configuration

```
Input: Time length $T$, Spatial dimensions $H \times W$
Output: Camera parameters $E$, $K$, projection matrices
procedure InitializeCamera
    $f_x \leftarrow T/2$                                  ▷ Temporal focal length
    $f_y \leftarrow (H \cdot W)/2$                        ▷ Spatial focal length
    FovX $\leftarrow 2\arctan(T/(2f_x))$                  ▷ Field-of-view in time
    FovY $\leftarrow 2\arctan((H \cdot W)/(2f_y))$        ▷ Field-of-view in space
    $\mathbf{E} \leftarrow \begin{bmatrix} f_x & 0 & f_x \\ 0 & f_y & f_y \\ 0 & 0 & 1 \end{bmatrix}$   ▷ Intrinsic matrix
    $\mathbf{K} \leftarrow \mathbf{I}_4$                   ▷ Extrinsic matrix (identity)
    $\mathbf{M}_{\text{proj}} \leftarrow$ getProjectionMatrix(FovX, FovY)   ▷ 3D-to-2D projection
    $\mathbf{M}_{\text{view}} \leftarrow$ getWorld2View($\mathbf{I}_3, \mathbf{0}$)   ▷ World-to-camera transform
end procedure
```

The Details of Gaussian Adapter. Our framework employs a learnable 4D Gaussian adapter G_θ to predict dynamic neural primitives that encode facial geometry, appearance, and motion. Gaussian adapter establishes a differentiable mapping between raw video sequences and structured 4D Gaussian maps $\mathbf{M}_{gs} \in \mathbb{R}^{22 \times N \times T}$, where N denotes spatial resolution and T temporal duration. Each Gaussian primitive contains 22 parameters organized into three functional groups:

1. Appearance Attributes.

- $\mathbf{d}_r \in \mathbb{R}^1$: Radial depth from camera plane

- $\mathbf{v}_s \in \mathbb{R}^3$: Specular reflection coefficients (Fresnel term)

- $\mathbf{v}_d \in \mathbb{R}^3$: Diffuse albedo (Lambertian component)

- $\mathbf{v}_n \in \mathbb{R}^3$: Motion-induced noise residuals

The final perceived color combines these components through:

$$\mathbf{c} = \underbrace{\mathbf{v}_s}_{\text{Specular}} + \underbrace{\mathbf{v}_d}_{\text{Phisological Signals}} + \underbrace{\mathbf{v}_n}_{\text{Motion Noise}}$$

2. Geometric Properties. - $\mathbf{a} \in \mathbb{R}^1$: Alpha transparency (density modulation)

- $\mathbf{s} \in \mathbb{R}^3$: Anisotropic scaling factors

- $\mathbf{r} \in \mathbb{R}^4$: Rotation quaternion (3D orientation)

3. Motion Dynamics.

- $[\Delta h, \Delta w, \Delta s] \in \mathbb{R}^5$: Spatiotemporal displacement field

- $\Delta h, \Delta w$: Spatial translation in image plane

- Δs: Scale evolution over time

4. Differentiable Rendering Formulation. The 3D position $\mathbf{p} \in \mathbb{R}^3$ of each Gaussian primitive is computed through perspective projection:

$$\mathbf{p} = \mathbf{KE} \begin{bmatrix} u \cdot d \\ v \cdot d \\ d \\ 1 \end{bmatrix},$$

$$\text{where} \begin{cases} (u, v) \in [0, 1]^2 : \text{UV coordinates} \\ d = \mathbf{d}_r : \text{Depth} \\ \mathbf{K}, \mathbf{E} : \text{Camera matrices} \end{cases}$$

This formulation ensures depth-aware preservation of facial topology during projection.

5. Activation Constraints.

Following [3], we employ specialized activation functions:

$$\begin{aligned}
\mathcal{F}_{\text{rgb}}(\mathbf{c}) &= \text{Sigmoid}(\mathbf{c}) \quad \text{(Color)}\\
\mathcal{F}_{\alpha}(\mathbf{a}) &= \text{Softplus}(\mathbf{a}) \quad \text{(Opacity)}\\
\mathcal{F}_{\text{scale}}(\mathbf{s}) &= e^{\zeta(\mathbf{s}-1)} \quad \text{(Scale)}\\
\mathcal{F}_{\text{rot}}(\mathbf{r}) &= \text{normalize}(\mathbf{r}) \quad \text{(Rotation)}
\end{aligned}$$

where ζ denotes a learnable curvature parameter. These constraints guarantee physically plausible gradient propagation during optimization.

6. Temporal Coherence The motion flow parameters $[\Delta h, \Delta w, \Delta s]$ enable explicit modeling of facial dynamics through:

$$\mathbf{p}_t = \mathbf{p}_{t-1} + \mathbf{J}^{-1}[\Delta h, \Delta w, \Delta s]^\top$$

where $\mathbf{J}$ represents the Jacobian of previous frame's projection. This differential formulation ensures temporal smoothness while accommodating non-rigid deformations.

Method	Training Time (h)	Model (ms)	MAE
DeepPhy [1]	40	400	11.0
NEST-rPPG [2]	8	8	4.76
PhysioGaussian	14	8	4.22

Table 1. Summary of Methods with Training Time, Model Inference Time, and MAE on VIPL dataset.

2. The Details of Experiment

Training and inference efficiency. We conducted a training evaluations using the VIPL dataset, focusing on the training time, inference time, and performance of three methodologies: DeepPhy, NEST-rPPG, and PhysioGaussian. The comparison of the methodologies is summarized in Table 1, which presents key metrics for each method, including their training times, inference times, and Mean Absolute Errors (MAE).

From the data presented in Table 1, it is evident that DeepPhy requires significantly more training time (40 hours) and exhibits a high inference time of 400 ms, resulting in a Mean Absolute Error (MAE) of 11.0. DeepPhy, which employs a video-based approach, displayed suboptimal performance in both training duration and inference efficiency. This inefficiency underscores the limitations of video-based methods in real-time applications. In contrast, both NEST-rPPG and PhysioGaussian demonstrated superior efficiency and effectiveness, leading to our decision to select STMap as the network input. Notably, PhysioGaussian simplifies the inference process by omitting the 4D Gaussian Adapter, resulting in a network structure that aligns perfectly with that of NEST-rPPG. However, it is important to highlight that PhysioGaussian requires simultaneous reconstruction during the training phase, consequently extending its training duration compared to NEST-rPPG. Nonetheless, given the relatively lightweight nature of the tasks involved, this additional training time is often deemed negligible in practical applications.

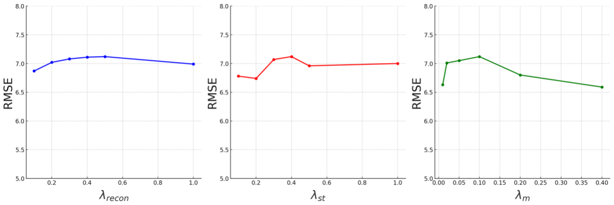

Figure 1. Effect of λ_{rec}, λ_{st}, and λ_m on VIPL-HR.

Effect of λ_{rec}, λ_{st}, and λ_m. Ablation studies on three balancing parameters (λ_{rec}, λ_{st}, and λ_m) reveal distinct functional roles (Fig. 1). The reconstruction constraint λ_{rec} shows minimal RMSE fluctuation (±0.2bpm), indicating limited influence of specular component (V_s) decoupling. In contrast, spatiotemporal consistency weight λ_{st} and motion noise constraint λ_m demonstrate critical importance - optimal performance emerges at $\lambda_{st} = 0.2$ and $\lambda_m = 0.01$. The λ_{st} enforces physiological continuity through V_d's spatiotemporal coherence, while λ_m isolates motion artifacts in V_n. Their synergy directly enhances rPPG signal quality with significantly better performance than λ_{rec}, confirming the dominance of diffuse chroma/noise decoupling in BVP recovery.

References

[1] Weixuan Chen and Daniel McDuff. Deepphys: Video-based physiological measurement using convolutional attention networks. *european conference on computer vision*, 2018. 2

[2] Hao Lu, Zitong Yu, Xuesong Niu, and Ying-Cong Chen. Neuron structure modeling for generalizable remote physiological measurement. In *CVPR*, pages 18589–18599, 2023. 2

[3] J. Tang, Z. Chen, X. Chen, T. Wang, G. Zeng, and Z. Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024. 2