EUROGRAPHICS 2026 / B. Masia and J. Thies
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 2

Improving Facial Rig Semantics for Tracking and Retargeting: Supplementary Material

D. Omens[1,2] A. Thurman[1] J. Yu[2] and R. Fedkiw[1,2]

[1]Stanford University, USA
[2]Epic Games, USA

1. Related Work: Hierarchy of Disentanglement

The authors have found it helpful to classify prior work in digital faces into a *hierarchy of disentanglement*, emphasizing that retargeting problems live at the highest level where semantic intention is disentangled from geometry. The discussion is presented here for reference.

There is a plethora of work aimed at representing faces in digital worlds. Although the seminal efforts relied quite heavily on the ability of artists to sculpt, paint, and animate both virtual characters and digital doubles, the impact of data-driven approaches has increased significantly over the years. A number of works mostly aim primarily to *disentangle the camera view* from a virtual model of the subject of interest, rasterizing photorealistic images from novel views. The most notable, e.g. [GTZN21, GPL*21, ZAB*22, DWS*23, ZYW*23, XCL*24, XGGZ24, CSK*22, LSSS18, TCS*23, KQG*23, BSS*23, TRP*24, WYL*23, ZBT22a, SBL*23, YZM*24, BLW*24], utilize either NeRFs [MST*21] or Gaussian Splats [KKLD23]. See also [MSS*21].

Going one step further, much research has been aimed at geometry reconstruction, which additionally (i.e. in addition to disentangling the camera viewpoint) *disentangles three dimensional geometry from texture and lighting*. Many such works utilize a PCA representation, usually either a 3DMM [BV99, EST*20] or FLAME [LBB*17], for regularization. See e.g. [LAGP09, LKA*17, ZCL*23, TRT*24, HSA*24, ZBT22b, RFD*24, WZZ*24, DBB22, WBH*22]. Other works use anatomical constraints [BB14, WBGB16] or sculpted blend shapes [GVWT13] for regularization. Similar in spirit to our approach, [BBC*24] points out that geometric reconstruction results can be improved by personalizing the FLAME expression coefficients.

Independent of geometry, various works aim to additionally *disentangle texture from lighting*, see e.g. [BZH*23, TZK*17, DYX*19, DTA*21, DBA*21, DAT*23, LRF*23, ZHXQ24, HLX24, DGHG*24, SSS*24, ASY*24]. The high-end approaches utilize a light stage [DHT*00, GFT*11, SWH*17].

When retargeting, it is important to *disentangle semantic intention from geometry*. Although there is a plethora of computer vision work interested in semantic intention for the sake of scene interpretation, these works are rarely interested in the 3D reconstruction of facial geometry. Those who are focused on both mainly work in the field of computer graphics and make use of so-called facial animation rigs, see e.g. [OBP*12]. Notable work on facial animation rigs includes [BCGF19, DHT*00, ZSCS04, LWP10, CBE*15, YZF*23, BODO20, YZC*23, KDP*24, MLL*24, CEM*22]. Publically available options for facial animation software include MetaHumans in the Unreal Engine [Epi25] as well as tools suites in Maya [Aut25] and Blender [Ble25].

2. Differentiating the Tracker: Additional Details

2.1. Derivation of the reformulated Broyden's method

Let u be some instance of $\tilde{v}$ for a given I and ψ, so that u only varies with $\theta_{\mathcal{T}}$ during optimization. Given distinct values θ_α and θ_β with $\Delta\theta = \theta_\beta - \theta_\alpha$, an estimate to $\frac{\partial u}{\partial \theta}$ at θ_α can be updated using Broyden's method [Bro65]

$$\left(\frac{\partial u}{\partial \theta}\Big|_{\theta=\theta_\alpha}\right)^{new} = \left(\frac{\partial u}{\partial \theta}\Big|_{\theta=\theta_\alpha}\right)^{old} + \frac{1}{(\Delta\theta)^T\Delta\theta}\left(u(\theta_\beta) - u(\theta_\alpha) - \left(\frac{\partial u}{\partial \theta}\Big|_{\theta=\theta_\alpha}\right)^{old}\Delta\theta\right)(\Delta\theta)^T \tag{1}$$

so that

$$\left(\frac{\partial u}{\partial \theta}\Big|_{\theta=\theta_\alpha}\right)^{new}\Delta\theta = u(\theta_\beta) - u(\theta_\alpha) \tag{2}$$

is satisfied. Equation 2 shows that the estimate for the derivative satisfies a secant-type equation in the direction of $\Delta\theta$, as is typical for Broyden-style methods. In fact, letting $s = ||\Delta\theta||_2 = \sqrt{(\Delta\theta)^T\Delta\theta}$ allows Equations 1 and 2 to be written as

$$\left(\frac{\partial u}{\partial \theta}\Big|_{\theta=\theta_\alpha}\right)^{new} = \left(\frac{\partial u}{\partial \theta}\Big|_{\theta=\theta_\alpha}\right)^{old} + \left(\frac{u(\theta_\alpha + s\widehat{\Delta\theta}) - u(\theta_\alpha)}{s} - \left(\frac{\partial u}{\partial \theta}\Big|_{\theta=\theta_\alpha}\right)^{old}\widehat{\Delta\theta}\right)\widehat{\Delta\theta}^T \tag{3}$$

$$\left(\frac{\partial u}{\partial \theta}\bigg|_{\theta=\theta_\alpha}\right)^{new} \widehat{\Delta\theta} = \frac{u(\theta_\alpha + s\widehat{\Delta\theta}) - u(\theta_\alpha)}{s} \tag{4}$$

where $\widehat{\Delta\theta}$ is a unit vector in the direction of $\Delta\theta$. Equations 3 and 4 illustrate that Equation 1 is updating $\frac{\partial u}{\partial \theta}$ to be consistent with a finite difference estimate of $\frac{\partial u}{\partial \theta}$ in the direction $\widehat{\Delta\theta}$.

The initial guess for $\frac{\partial u}{\partial \theta}$ in the first iteration of Equation 3 can be set to be the final value obtained for $\frac{\partial u}{\partial \theta}$ when iterating Equation 3 in the prior optimization step. This warm start bears resemblance to Broyden's method, and it greatly accelerated convergence in all of our tests. In the first optimization iteration where there is no prior estimate for $\frac{\partial u}{\partial \theta}$, an initial guess of zero can be used. This differs from Broyden's method, which uses the identity because the estimate is typically used in conjunction with matrix inversion.

2.2. Using Broyden's Method to estimate $\frac{\partial \tilde{v}}{\partial \theta}$

Although one could assume that the tracker produces a $\hat{v}$ very close to v making $\tilde{v}$ small enough to remove it from Equation 5, subsequently removing its derivative from Equation 7, it is also possible to estimate $\frac{\partial \tilde{v}}{\partial \theta}$. In order to estimate $\frac{\partial \tilde{v}}{\partial \theta}$, we utilize the finite difference update at the heart of Broyden's method [Bro65], which is also the motivation for many other optimization schemes including SR1 [NW06], DFP [Dav91, FP63], BFGS [Fle87], and L-BFGS [LN89].

Unlike the typical approach which compiles a running estimate of $\frac{\partial u}{\partial \theta}$ as the optimization steps through parameter space, we use Equation 3 in order to construct an estimate of $\frac{\partial u}{\partial \theta}$ at θ_α. This is accomplished by using Equation 3 repeatedly with different $\widehat{\Delta\theta}$ directions. The various $\widehat{\Delta\theta}$ can be chosen randomly, importance sampled, etc., noting that the estimate of $\frac{\partial u}{\partial \theta}$ is merely used to aid in the choice of the search direction; thus, this approach is perhaps best described as a predictor-corrector method similar in spirit to Nesterov or second-order Runge-Kutta. Finally, note that each $\widehat{\Delta\theta}$ direction can use a varying s in order to ascertain a reasonable finite difference approximation that avoids round-off error while minimizing truncation error.

2.3. $\tilde{v}$ versus $\hat{v}$

Differentiating

$$\hat{v}(I;\theta_\mathcal{T},\psi) = v(I) + \tilde{v}(I;\theta_\mathcal{T},\psi) = \mathcal{R}(\mathcal{T}(I;\theta_\mathcal{T},\psi);\theta_\mathcal{T}) \tag{5}$$

with respect to θ leads directly to

$$\frac{\partial \tilde{v}(I;\theta,\psi)}{\partial \theta}\bigg|_{\theta=\theta_\mathcal{T}} = \frac{\partial \mathcal{R}(c;\theta_\mathcal{T})}{\partial c}\bigg|_{c=\mathcal{T}(I;\theta_\mathcal{T},\psi)} \frac{\partial \mathcal{T}(I;\theta,\psi)}{\partial \theta}\bigg|_{\theta=\theta_\mathcal{T}} + \frac{\partial \mathcal{R}(\mathcal{T}(I;\theta_\mathcal{T},\psi);\theta)}{\partial \theta}\bigg|_{\theta=\theta_\mathcal{T}} \tag{6}$$

and thus

$$\frac{\partial \mathcal{R}(c;\theta_\mathcal{T})}{\partial c}\bigg|_{c=\mathcal{T}(I;\theta_\mathcal{T},\psi)} \frac{\partial \mathcal{T}(I;\theta,\psi)}{\partial \theta}\bigg|_{\theta=\theta_\mathcal{T}} = -\frac{\partial \mathcal{R}(\mathcal{T}(I;\theta_\mathcal{T},\psi);\theta)}{\partial \theta}\bigg|_{\theta=\theta_\mathcal{T}} + \frac{\partial \tilde{v}(I;\theta,\psi)}{\partial \theta}\bigg|_{\theta=\theta_\mathcal{T}} \tag{7}$$

as an implicit equation for $\frac{\partial \mathcal{T}}{\partial \theta}$. In our derivations of the reformulated Broyden's method, note that that u cannot actually be an instance of $\tilde{v}$, since $\tilde{v}$ represents unknown error; however, letting u be an instance of $\hat{v}$, which is equivalent to the rig output (Equation 5), still leads to an estimate of $\frac{\partial \tilde{v}}{\partial \theta}$ since the exact solution v does not vary with θ. Therefore,

$$\frac{\partial \mathcal{R}(c;\theta_\mathcal{T})}{\partial c}\bigg|_{c=\mathcal{T}(I;\theta_\mathcal{T},\psi)} \frac{\partial \mathcal{T}(I;\theta,\psi)}{\partial \theta}\bigg|_{\theta=\theta_\mathcal{T}} = -\frac{\partial \mathcal{R}(\mathcal{T}(I;\theta_\mathcal{T},\psi);\theta)}{\partial \theta}\bigg|_{\theta=\theta_\mathcal{T}} + \frac{\partial \hat{v}(I;\theta,\psi)}{\partial \theta}\bigg|_{\theta=\theta_\mathcal{T}} \tag{8}$$

is used as a replacement for Equation 7 in the method.

3. Linear Examples: Additional Details

3.1. Assuming linearity of the tracker

Assuming linearity of the tracker according to $c = B(\psi)v$ led to the notion of a tracker that solves a linear system $\hat{A}c = v$. Letting $\hat{A}^{-1}v$ represent the result obtained by the tracker when solving this linear system leads to

$$\begin{aligned}\min_{\theta_\mathcal{T}} \sum_k \Big(&\gamma_1 ||\hat{A}^{-1}(\theta_\mathcal{T})v_k - c_k||_2^2 + \\ &\gamma_2 ||A(\theta_\mathcal{R})\hat{A}^{-1}(\theta_\mathcal{T})v_k - v_k||_2^2 + \\ &\gamma_3 ||\hat{A}(\theta_\mathcal{T})\hat{A}^{-1}(\theta_\mathcal{T})v_k - v_k||_2^2\Big) + \\ &\gamma_\epsilon ||\theta_\mathcal{T} - \theta_\mathcal{R}||_2^2\end{aligned} \tag{9}$$

in place of

$$\begin{aligned}\min_{\theta_\mathcal{T}} \sum_k \Big(&\gamma_1 ||\mathcal{T}(v_k;\theta_\mathcal{T}) - c_k||_2^2 + \\ &\gamma_2 ||A(\theta_\mathcal{R})\mathcal{T}(v_k;\theta_\mathcal{T}) - v_k||_2^2 + \\ &\gamma_3 ||\hat{A}(\theta_\mathcal{T})\mathcal{T}(v_k;\theta_\mathcal{T}) - v_k||_2^2\Big) + \\ &\gamma_\epsilon ||\theta_\mathcal{T} - \theta_\mathcal{R}||_2^2\end{aligned} \tag{10}$$

Note that errors in the tracker solve, regularization, etc. typically cause $\hat{A}$ and $\hat{A}^{-1}$ to not necessarily cancel. In the space spanned by the v_k, the second term aims to make $\hat{A}^{-1}$ the right inverse of A (driving $\theta_\mathcal{T}$ to $\theta_\mathcal{R}$ in a subspace), while the third term aims to make $\hat{A}$ and $\hat{A}^{-1}$ the left/right inverses of each other. When $A(\theta_\mathcal{R})c_k = v_k$, the second term is merely a scaling of the first term. When $A(\theta_\mathcal{R})c_k \neq v_k$, which arises when an inconsistent set of (c_k, v_k) pairs makes $C^T A^T(\theta_\mathcal{R}) = V^T$ overdetermined, the second term provides information beyond that of the first term, penalizing the solution to better match the geometry.

Solving Equation 10 requires the computation of $\frac{\partial \mathcal{T}}{\partial \theta}$ for each of the k terms in the sum. In this linearized scenario, Equations 5 and 8 become

$$\hat{v}(v;\theta_\mathcal{T}) = \hat{A}(\theta_\mathcal{T})\mathcal{T}(v;\theta_\mathcal{T}) \tag{11}$$

$$\hat{A}(\theta_\mathcal{T})\frac{\partial \mathcal{T}(v;\theta_\mathcal{T})}{\partial \theta_\mathcal{T}} = -\begin{bmatrix}\mathcal{T}(v;\theta_\mathcal{T})^T & 0_{1x3} & 0_{1x3} \\ 0_{1x3} & \mathcal{T}(v;\theta_\mathcal{T})^T & 0_{1x3} \\ 0_{1x3} & 0_{1x3} & \mathcal{T}(v;\theta_\mathcal{T})^T\end{bmatrix} + \frac{\partial \hat{v}(v;\theta_\mathcal{T})}{\partial \theta_\mathcal{T}} \tag{12}$$

where $\mathcal{T}(v;\theta_\mathcal{T})$ is found by solving $\hat{A}c = v$. This solve can be accomplished via the inverse (when it exists), least squares (when $\hat{A}$ is full rank), or the pseudo-inverse (to obtain the minimum norm solution, when $\hat{A}$ is not full rank); optionally, the equations can be augmented with Levenberg-Marquardt regularization and solved via the normal equations. In order to estimate $\frac{\partial \hat{v}}{\partial \theta}$, Equation 3 is iterated for various $\widehat{\Delta\theta}$ directions; afterwards, Equation 12 can be solved independently for each column of $\frac{\partial \mathcal{T}}{\partial \theta}$.

3.2. Analytic solutions for loss terms

For completeness, the derivatives of the four terms in Equation 10 are

$$\frac{\partial \mathcal{L}_{\gamma_1}}{\partial \theta_\mathcal{T}} = 2\gamma_1 \sum_k (\mathcal{T}(v_k;\theta_\mathcal{T}) - c_k)^T \frac{\partial \mathcal{T}(v_k;\theta_\mathcal{T})}{\partial \theta_\mathcal{T}} \tag{13}$$

$$\frac{\partial \mathcal{L}_{\gamma_2}}{\partial \theta_\mathcal{T}} = 2\gamma_2 \sum_k (A(\theta_\mathcal{R})\mathcal{T}(v_k;\theta_\mathcal{T}) - v_k)^T A(\theta_\mathcal{R}) \frac{\partial \mathcal{T}(v_k;\theta_\mathcal{T})}{\partial \theta_\mathcal{T}} \tag{14}$$

$$\frac{\partial \mathcal{L}_{\gamma_3}}{\partial \theta_\mathcal{T}} = 2\gamma_3 \sum_k (\hat{A}(\theta_\mathcal{T})\mathcal{T}(v_k;\theta_\mathcal{T}) - v_k)^T \left(\hat{A}(\theta_\mathcal{T}) \frac{\partial \mathcal{T}(v_k;\theta_\mathcal{T})}{\partial \theta_\mathcal{T}} + \begin{bmatrix} \mathcal{T}(v_k;\theta_\mathcal{T})^T & 0_{1x3} & 0_{1x3} \\ 0_{1x3} & \mathcal{T}(v_k;\theta_\mathcal{T})^T & 0_{1x3} \\ 0_{1x3} & 0_{1x3} & \mathcal{T}(v_k;\theta_\mathcal{T})^T \end{bmatrix} \right) \tag{15}$$

$$\frac{\partial \mathcal{L}_{\gamma_\epsilon}}{\partial \theta_\mathcal{T}} = 2\gamma_\epsilon (\theta_\mathcal{T} - \theta_\mathcal{R})^T \tag{16}$$

where the term in parentheses in Equation 15 would be replaced by

$$\frac{\partial \hat{A}}{\partial \theta_\mathcal{T}} = \left.\frac{\partial \hat{A}(c;\theta_\mathcal{T})}{\partial c}\right|_{c=\mathcal{T}(v_k;\theta_\mathcal{T})} \frac{\partial \mathcal{T}(v_k;\theta_\mathcal{T})}{\partial \theta_\mathcal{T}} + \left.\frac{\partial \hat{A}(c;\theta_\mathcal{T})}{\partial \theta_\mathcal{T}}\right|_{c=\mathcal{T}(v_k;\theta_\mathcal{T})} \tag{17}$$

when $\hat{A}(\mathcal{T}(v_k;\theta_\mathcal{T});\theta_\mathcal{T})$ is nonlinear.

3.3. Avoiding ∞ times 0 singularities

Consider a single pair $(c_1, v_1) = (1, -1)$ where $\hat{A} = [-1]$. For now, consider only the γ_1 term. Starting with an initial guess of $\hat{A}_0 = 1$, the tracker solves $\hat{A}_0 C_0 = V$ to obtain $C_0 = [-1]$; then, the optimization attempts to drive C from $[-1]$ to $[1]$ by sending $\hat{A} \to [\infty]$ to obtain $C \to [0]$. See Figure 1. Unfortunately, the origin is a non-removable singularity, and there is no mechanism to cross over the origin changing the entry in C from negative to positive while flipping the entry in $\hat{A}$ from ∞ to $-\infty$.

Next, consider

$$\hat{A} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, C = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}, V = \begin{bmatrix} -1 & -1 \\ -2 & -1 \end{bmatrix} \tag{18}$$

where $\hat{A}C = V$. When starting with an initial guess of $\hat{A}_0 = I$, the optimization suffers from the same issue as was discussed in the prior example. See Figure 2.

Changing the initial guess to

$$\hat{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \tag{19}$$

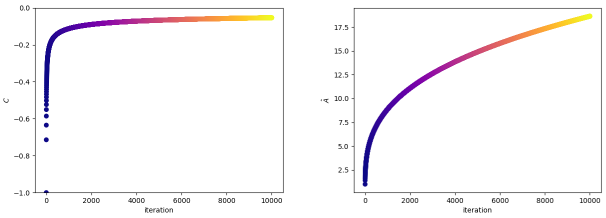

Figure 1: *As $C \to [0]$, $\hat{A} \to [\infty]$.*

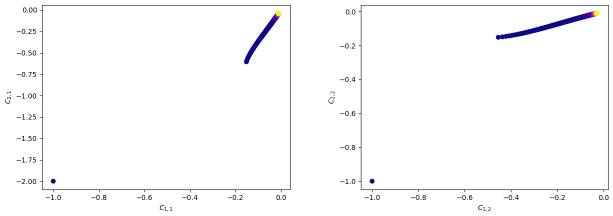

Figure 2: *Starting with $\hat{A}_0 = I$ leads to $C_0 = V$. The left/right subfigure shows the first/second column of C with each row entry on a separate axis. As the iteration proceeds, the results are color-coded from blue to yellow. All four entries are driven to zero; likewise, all four entries of $\hat{A}$ blow up.*

achieves the expected results, since the origin is avoided. See Figure 3. Note that perturbing the initial guess is a viable strategy because the optimization is done offline (and only once) in order to determine a suitable tracking rig. Moreover, using $\theta_\mathcal{R}$ as an initial guess for $\theta_\mathcal{T}$ is likely a good strategy, especially when the tracking rig does not need to be perturbed too much from the animation rig.

3.4. Solver Efficacy

The data values we use for our example are

$$C = \begin{bmatrix} 1 & 2 & 3 & 1 \\ 2 & -1 & 1 & 1 \\ 3 & -1 & -2 & 1 \end{bmatrix}, V = \begin{bmatrix} -1 & -2 & -3 & -1 \\ 4 & -2 & 2 & 2 \\ -2 & \frac{2}{3} & \frac{4}{3} & -\frac{2}{3} \end{bmatrix} \tag{20}$$

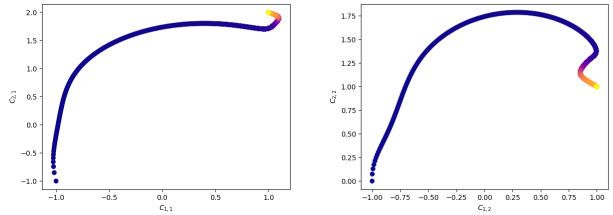

Figure 3: *Changing the initial guess allows both C and $\hat{A}$ to converge correctly.*

	$\mathcal{L}_D$	$\mathcal{L}_{\gamma_1}+\mathcal{L}_{\gamma_2}+\mathcal{L}_{\gamma_3}$	$\|\|\hat{A}-A\|\|_F^2$
Direct	4.58E−12	4.85E−12	4.26E−13
γ_1 only	2.05E−8	2.58E−8	1.31E−8
γ_2 only	1.54E−9	4.15E−9	9.82E−10

Table 1: *Comparison against the direct method using all four columns of C and (the un-perturbed) V.*

	$\mathcal{L}_D$	$\mathcal{L}_{\gamma_1}$	$\mathcal{L}_{\gamma_2}$	$\|\|\hat{A}-\hat{A}_4\|\|_F^2$	$\|\|\hat{A}-A\|\|_F^2$
Direct	6.60E−4	3.47E−4	2.55E−3	0	5.57E−4
γ_1 only	6.60E−4	3.47E−4	2.54E−3	9.25E−10	5.57E−4
γ_2 only	3.21E−3	2.38E−3	5.65E−9	5.58E−4	3.63E−9

Table 2: *Comparison against the direct method starting with $\hat{A}=I$ and using the perturbed V.*

consistent with a rig

$$A=\begin{bmatrix}-1 & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & -\frac{2}{3}\end{bmatrix} \tag{21}$$

Our perturbed version of V was

$$\hat{V}=\begin{bmatrix}-0.990 & -2.00 & -2.97 & -1.00\\ 3.97 & -1.98 & 1.98 & 2.01\\ -1.98 & 0.667 & 1.33 & -0.667\end{bmatrix} \tag{22}$$

The results in Table 1 were obtained using all four columns of C and (the un-perturbed) V.

Using the perturbed version of V, the results obtained starting with $\hat{A}=I$ are shown in Table 2. Since the pairs over-determine $\hat{A}$, the direct method cannot achieve a small residual and the γ_1 term is bounded from below; however, they both result in the correct least squares solution (specified by $\hat{A}_4$ in the table). Using only the first three columns of C and $\hat{V}$ allows both the direct method and the γ_1 term to achieve significantly smaller errors (as expected) since $\hat{A}$ is no longer over-determined. The γ_2 drives $\hat{A}\rightarrow A$ as expected.

3.5. Improving the search direction via $\frac{\partial\hat{v}}{\partial\theta_{\mathcal{T}}}$

Consider a tracker $\mathcal{T}(v;\theta_{\mathcal{T}})=\hat{A}^{\dagger}v$ where $\hat{A}^{\dagger}$ depends on the method being used to solve $\hat{A}c=v$. Next, perturb the output of the tracker to

$$\mathcal{T}_1(v;\theta_{\mathcal{T}})=\hat{A}^{\dagger}v+\tilde{c} \tag{23}$$

where $\tilde{c}$ captures the fact that trackers often (intentionally, for robustness) do not correctly/precisely invert the rig. Equation 5 becomes

$$\hat{v}(v;\theta_{\mathcal{T}})=v+\tilde{v}(v;\theta_{\mathcal{T}})=\hat{A}(\theta_{\mathcal{T}})\left(\hat{A}^{\dagger}(\theta_{\mathcal{T}})v+\tilde{c}(v)\right) \tag{24}$$

where $\tilde{v}\neq 0$. When $\hat{A}$ happens to be invertible,

$$\tilde{v}(v;\theta_{\mathcal{T}})=\hat{A}(\theta_{\mathcal{T}})\tilde{c}(v) \tag{25}$$

	$\|\|V-\hat{A}C+\hat{A}\tilde{C}\|\|_F^2$	iters
γ_1 only	3.86E−8	13868
γ_1 & $\frac{\partial\hat{v}}{\partial\theta_{\mathcal{T}}}$	3.82E−8	12960

Table 3: *Results obtained with and without using $\frac{\partial\hat{v}}{\partial\theta_{\mathcal{T}}}$ to improve the search direction.*

$$\frac{\partial\hat{v}(v;\theta_{\mathcal{T}})}{\partial\theta_{\mathcal{T}}}=\frac{\partial\tilde{v}(v;\theta_{\mathcal{T}})}{\partial\theta_{\mathcal{T}}}=\begin{bmatrix}\tilde{c}(v)^T & 0_{1x3} & 0_{1x3}\\ 0_{1x3} & \tilde{c}(v)^T & 0_{1x3}\\ 0_{1x3} & 0_{1x3} & \tilde{c}(v)^T\end{bmatrix} \tag{26}$$

leading to

$$\mathcal{L}_{\hat{v}'}=\sum_k\left|\left|\frac{\partial\hat{v}_k(v_k;\theta_{\mathcal{T}})}{\partial\theta_{\mathcal{T}}}-\begin{bmatrix}\tilde{c}_k(v_k)^T & 0_{1x3} & 0_{1x3}\\ 0_{1x3} & \tilde{c}_k(v_k)^T & 0_{1x3}\\ 0_{1x3} & 0_{1x3} & \tilde{c}_k(v_k)^T\end{bmatrix}\right|\right|_F^2 \tag{27}$$

as a way of evaluating the efficacy of approximations to $\frac{\partial\hat{v}}{\partial\theta_{\mathcal{T}}}$.

Plugging Equation 23 into the γ_1 term in Equation 10 and stacking columns leads to $\hat{A}^{\dagger}V+\tilde{C}-C=0$ or $V-\hat{A}C+\hat{A}\tilde{C}=0$ when $\hat{A}$ happens to be invertible; thus, minimizing the γ_1 term leads to a small value for $V-\hat{A}C+\hat{A}\tilde{C}$ instead of a small value for $V-\hat{A}C$. Table 3 shows the results obtained with and without using $\frac{\partial\hat{v}}{\partial\theta_{\mathcal{T}}}$ to improve the search direction. In this simple example, auto-diff can be used to compute $\frac{\partial\hat{v}}{\partial\theta_{\mathcal{T}}}$. The average $\mathcal{L}_{\hat{v}'}$ error obtained when using auto-diff for $\frac{\partial\hat{v}}{\partial\theta_{\mathcal{T}}}$ was 6.44E−10; for the sake of comparison, the average value of $\sum_k\left|\left|\frac{\partial\hat{v}_k}{\partial\theta_{\mathcal{T}}}\right|\right|_F^2$ was 7.05E−3. Only the first three columns of C and V were used, $\tilde{C}=A^{-1}(\hat{V}-V)$ was chosen so that the perturbations in C resemble those used for V, and the initial guess was $\hat{A}=I$. The γ_2 term gave similar results, as expected since $AC=V$ here.

Next, consider perturbing the output of the tracker to

$$\mathcal{T}_2(v;\theta_{\mathcal{T}})=\hat{A}^{\dagger}v+\hat{A}(\theta_{\mathcal{T}})\tilde{c}(v) \tag{28}$$

instead of Equation 23. Equation 5 becomes

$$\hat{v}(v;\theta_{\mathcal{T}})=v+\tilde{v}(v;\theta_{\mathcal{T}})=\hat{A}(\theta_{\mathcal{T}})(\hat{A}^{\dagger}(\theta_{\mathcal{T}})v+\hat{A}(\theta_{\mathcal{T}})\tilde{c}(v)) \tag{29}$$

and

$$\tilde{v}(v;\theta_{\mathcal{T}})=\hat{A}(\theta_{\mathcal{T}})\hat{A}(\theta_{\mathcal{T}})\tilde{c}(v) \tag{30}$$

$$\begin{aligned}\frac{\partial\hat{v}(v;\theta_{\mathcal{T}})}{\partial\theta_{\mathcal{T}}}=&\hat{A}(\theta_{\mathcal{T}})\begin{bmatrix}\tilde{c}(v)^T & 0_{1x3} & 0_{1x3}\\ 0_{1x3} & \tilde{c}(v)^T & 0_{1x3}\\ 0_{1x3} & 0_{1x3} & \tilde{c}(v)^T\end{bmatrix}+\\ &\begin{bmatrix}\tilde{c}(v)^T\hat{A}^T(\theta_{\mathcal{T}}) & 0_{1x3} & 0_{1x3}\\ 0_{1x3} & \tilde{c}(v)^T\hat{A}^T(\theta_{\mathcal{T}}) & 0_{1x3}\\ 0_{1x3} & 0_{1x3} & \tilde{c}(v)^T\hat{A}^T(\theta_{\mathcal{T}})\end{bmatrix}\end{aligned} \tag{31}$$

replaces Equations 25 and 26 when $\hat{A}$ happens to be invertible. Equation 31 can be used to define a new $\mathcal{L}_{\hat{v}'}$ that replaces Equation 27. Plugging Equation 28 into the γ_1 term in Equation 10 and stacking columns leads to the notion that minimizing the γ_1 term should lead to a small value of $V-\hat{A}C+\hat{A}^2\tilde{C}$, assuming that $\hat{A}$ happens to be invertible. Table 6 shows the results obtained with and

	$\|\|V-\hat{A}C+\hat{A}^2\tilde{C}\|\|_F^2$	iters
γ_1 only	3.82E−8	29083
γ_1 & $\frac{\partial\hat{v}}{\partial\theta_\mathcal{T}}$	3.79E−8	5329

Table 4: *Results obtained with and without using $\frac{\partial\hat{v}}{\partial\theta_\mathcal{T}}$ to improve the search direction, using the perturbed tracker.*

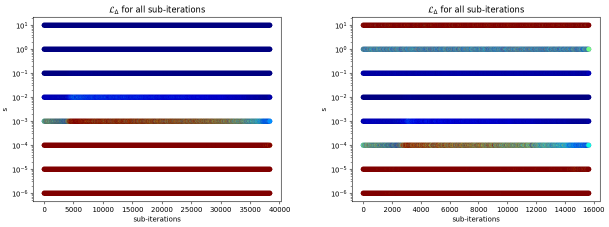

Figure 4: *Color-coded from lower error (blue) to higher error (red). Left: $\mathcal{T}_1$. Right: $\mathcal{T}_2$.*

without using $\frac{\partial\hat{v}}{\partial\theta_\mathcal{T}}$ to improve the search direction. Once again, this example is simple enough to use auto-diff. The average $\mathcal{L}_{\hat{v}'}$ error obtained when using auto-diff for $\frac{\partial\hat{v}}{\partial\theta_\mathcal{T}}$ was 2.54E−10; for the sake of comparison, the average value of $\sum_k \left|\left|\frac{\partial\hat{v}_k}{\partial\theta_\mathcal{T}}\right|\right|_F^2$ was 1.91E−1.

3.6. Estimating $\frac{\partial\hat{v}}{\partial\theta_\mathcal{T}}$

A general black-box tracker is not necessarily amenable to auto-diff; in such a scenario, $\frac{\partial\hat{v}}{\partial\theta_\mathcal{T}}$ would need to be estimated via Equation 3. In order to demonstrate the efficacy of Equation 3, consider minimizing the γ_1 term using either $\mathcal{T}_1$ from Equation 23 or $\mathcal{T}_2$ from Equation 24 along with the analytic values for $\frac{\partial\hat{v}}{\partial\theta_\mathcal{T}}$ given in Equations 26 and 31. After about 13000 optimization steps, the analytic solution for $\frac{\partial\hat{v}_k}{\partial\theta_\mathcal{T}}$ is used about 39000 times since C and V have three columns.

In order to ascertain the sensitivity of the parameter s in the finite difference approach, an approximation to $\frac{\partial\hat{v}_k}{\partial\theta_\mathcal{T}}$ was computed via one iteration of Equation 3 using a randomly chosen $\widehat{\Delta\theta}$ direction and the following values of s: 1E−7, 1E−6, 1E−5, 1E−4, 1E−3, 1E−2, 1E−1, 1E0, 1E1, 1E2. The finite difference approximations for any two values of s can be compared via a Frobenius norm. Let $\mathcal{L}_\Delta(s;\theta_\mathcal{T},\widehat{\Delta\theta})$ be the sum of the Frobenius norms obtained by comparing s to both its next higher and next lower values; then, $\mathcal{L}_\Delta$ indicates the relative sensitivity of the finite difference approximations to perturbations in s. Figure 4 shows the results obtained on all 39000 sub-iterations. The red regions at the bottom of the figures represent round-off errors, and the red region at the top of the right figure represents truncation error (as expected, see e.g. [Gea71]). There is no red region at the top of the left figure, since $\frac{\partial\hat{v}_k}{\partial\theta_\mathcal{T}}$ is constant (see Equation 26). The large blue region illustrates the ease at which s can be chosen.

Next, consider the same tests with the analytic solution replaced by various approximations to the $\frac{\partial\hat{v}_k}{\partial\theta_\mathcal{T}}$. For the finite difference approximations, s was chosen to minimize $\mathcal{L}_\Delta$. Tables 5 and 6 show the results obtained for $\mathcal{T}_1$ and $\mathcal{T}_2$ respectively using either a number of randomly-chosen $\widehat{\Delta\theta}$ directions, auto-diff, or a single $\widehat{\Delta\theta}$ chosen in the direction of steepest descent. The third column of the tables averages the error (see Equations 27 and 31) over all iterations; for the sake of comparison, the second column averages the solution. As expected, increasing the number of randomly chosen $\widehat{\Delta\theta}$ improves the estimate of $\frac{\partial\hat{v}_k}{\partial\theta_\mathcal{T}}$; however, improved estimates of $\frac{\partial\hat{v}_k}{\partial\theta_\mathcal{T}}$ did not tend to improve convergence. On the other hand, choosing $\widehat{\Delta\theta}$ in the steepest descent direction did improve convergence. This makes sense from the viewpoint of a predictor-corrector approach, even though $\frac{\partial\hat{v}_k}{\partial\theta_\mathcal{T}}$ is poorly estimated as compared to the other approaches. Unfortunately, iterating on steepest descent choices for $\widehat{\Delta\theta}$ did not further improve convergence.

$\mathcal{T}_1$	iters	$\mu\left(\sum_k \left\|\left\|\frac{\partial\hat{v}_k}{\partial\theta_\mathcal{T}}\right\|\right\|_F^2\right)$	$\mu(\mathcal{L}_{\hat{v}'})$
1 random	12781	7.04E−3	4.57E−6
10 random	12921	7.05E−3	2.86E−7
100 random	12964	7.05E−3	4.63E−8
auto-diff	12960	7.05E−3	1.74E−11
steepest descent	9089	2.09E−3	6.41E−3

Table 5: *Results obtained for $\mathcal{T}_1$ using either a number of randomly-chosen $\widehat{\Delta\theta}$ directions, auto-diff, or a single $\widehat{\Delta\theta}$ chosen in the direction of steepest descent.*

$\mathcal{T}_2$	iters	$\mu\left(\sum_k \left\|\left\|\frac{\partial\hat{v}_k}{\partial\theta_\mathcal{T}}\right\|\right\|_F^2\right)$	$\mu(\mathcal{L}_{\hat{v}'})$
1 random	4763	1.71E−1	3.50E−4
10 random	5307	1.93E−1	6.32E−6
100 random	5329	1.94E−1	1.24E−6
auto-diff	5329	1.94E−1	2.57E−10
steepest descent	4825	7.28E−2	1.93E−1

Table 6: *Results obtained for $\mathcal{T}_2$ using either a number of randomly-chosen $\widehat{\Delta\theta}$ directions, auto-diff, or a single $\widehat{\Delta\theta}$ chosen in the direction of steepest descent.*

4. Simon Says Animation Rig Creation: Additional Details

The nineteen expressions in our MetaHuman/FLAME expression set are “neutral”, “brows down”, “brows up”, “eyes wide”, “eyes close”, “nose wrinkle”, “cheek puff”, “teeth grimace”, “corner pull”, “mouth stretch”, “corner depress”, “lip press”, “pursed lips”, “mouth funnel”, “lip bite”, “jaw open”, “jaw open extreme”, “jaw left”, and “jaw right”. The nineteen expressions along with their associated animation controls are shown in Figure 5.

For our experiments, we used three additional combination regularization expressions directed by technical artists to help with lip sealing. Thsese are “jaw open lips together” (activating jaw and lips together), “jaw open lips purse” (activating jaw and lip purse), and

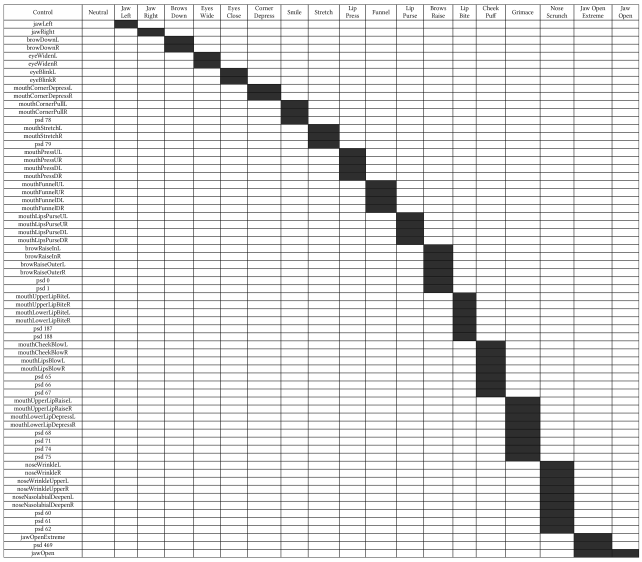

Figure 5: *For each expression, the associated column shows the active MetaHuman PSD rig controls.*

"B pop phoneme" (activating jaw, lip bite, lips together, purse, and stretch).

Each expression includes a handcrafted mask so that parts of the face that are not relevant to the expression do not adversely affect the result. For example, the mouth is commonly inadvertently moved during the "nose wrinkle" expression, but any movement of the map should not be calibrated into that expression.

4.1. Incorrect expressions

When the user struggles to make the correct expression (such as accidentally flexing eyebrows while smiling), the reconstructed geometry v will not correctly correspond to the expression specified by c. It is possible to roughly identify the additional controls that need to be activated in order to explain these spurious geometric deformations. Using the current best guess for the animation rig parameters $\theta_{\mathcal{R}}$, a tracker can be used to compute

$$c^{+} = c + (I_{n\times n} - \mathcal{H}(c))\,\mathcal{T}(I,\theta_{\mathcal{R}}) \tag{32}$$

where $\mathcal{H}(c)$ is a diagonal matrix of Heaviside functions $H(|c_i|)$. Equation 32 leaves the nonzero entries of c umodified while augmenting the other entries to agree with $\mathcal{T}(I,\theta_{\mathcal{R}})$. The newly added (c^{+},v) pairs help to prevent the inappropriate modifications to $\theta_{\mathcal{R}}$ that would have resulted from the (c,v) pairs that they replace when solving

$$\min_{\theta_{\mathcal{R}}} \sum_{k} ||\mathcal{R}(c_k;N,\theta_{\mathcal{R}}) - v_k||_2^2 \tag{33}$$

Alternatively, the geometry can be constrained to mask out regions of the face that should not deform for the given expression, replacing (c,v) pairs with (c,v^{-}) pairs.

The augmentation of c to c^{+} can be problematic when the added controls interfere with the primary expression under consideration; thus, it is desirable to leave interfering controls unmodified. An alternative strategy consists of constraining $\theta_{\mathcal{R}}$ so that the degrees of freedom irrelevant to c do not change. This is plausible when solving Equation 33 since $\mathcal{R}$ is typically a straightforward deterministic function; however, it can be daunting when solving Equation 34, since $\mathcal{T}$ contains both an ill-posed geometric reconstruction and an inverse problem (rig inversion).

When modifying c to c^{+} via Equation 32, the tracker fine-tuning equation

$$\begin{aligned}\min_{\theta_{\mathcal{T}}} \sum_{k} \Big(&\gamma_1 ||\mathcal{T}(I_k;\theta_{\mathcal{T}},\psi) - c_k||_2^2 + \\ &\gamma_2 ||\mathcal{R}(\mathcal{T}(I_k;\theta_{\mathcal{T}},\psi);\theta_{\mathcal{R}}) - v_k||_2^2\Big) + \\ &\gamma_\epsilon ||\theta_{\mathcal{T}} - \theta_{\mathcal{R}}||_2^2\end{aligned} \tag{34}$$

can be written as

$$\begin{aligned}\min_{\theta_{\mathcal{T}}} \sum_{k} \Big(&\gamma_1 ||\mathcal{T}(I_k;\theta_{\mathcal{T}}) - c_k^{+}||_2^2 + \\ &\gamma_2 ||\mathcal{R}(\mathcal{T}(I_k;\theta_{\mathcal{T}});\theta_{\mathcal{R}}) - v_k^{+}||_2^2\Big) + \\ &\gamma_\epsilon ||\theta_{\mathcal{T}} - \theta_{\mathcal{R}}||_2^2\end{aligned} \tag{35}$$

where $v^{+} = \mathcal{R}(c^{+};\theta_{\mathcal{R}})$, and omitting ϕ for brevity. Equations 5 and 8 can be written as

$$\hat{v}(I;\theta_{\mathcal{T}}) = \mathcal{R}(\mathcal{T}(I;\theta_{\mathcal{T}});\theta_{\mathcal{T}}) \tag{36a}$$

$$\begin{aligned}\frac{\partial \mathcal{R}(c;\theta_{\mathcal{T}})}{\partial c}\bigg|_{c=\mathcal{T}(I;\theta_{\mathcal{T}})} \frac{\partial \mathcal{T}(I;\theta)}{\partial \theta}\bigg|_{\theta=\theta_{\mathcal{T}}} = \\ -\frac{\partial \mathcal{R}(\mathcal{T}(I;\theta_{\mathcal{T}});\theta)}{\partial \theta}\bigg|_{\theta=\theta_{\mathcal{T}}} + \frac{\partial \hat{v}(I;\theta)}{\partial \theta}\bigg|_{\theta=\theta_{\mathcal{T}}}\end{aligned} \tag{36b}$$

in order to differentiate Equation 35. The pseudo-inverse of the normal equations can be used to solve for $\frac{\partial \mathcal{T}}{\partial \theta}$ in Equation 36b.

4.2. Experimental setup

For running the Simon-Says rig calibration process, we use the L-BFGS optimizer [LN89] PyTorch implementation for 4 iterations with `tolerance_grad` $= 1\text{E}{-7}$. We use a L2 squared penalty on the vertex distance to the target geometry with weight 1 for the MetaHuman and game character rig frameworks and weight 100 for the semantic FLAME framework (because of its small absolute size). We used a mean L2 squared penalty on the rig parameters with weight 1E1 for the MetaHuman rig framework, $1\text{E}{-1}$ for the game character rig framework, and 5E0 for the semantic FLAME rig framework.

5. Fine-tuning the Tracking Rig: Additional Details

5.1. MetaHuman framework

The MetaHuman Animator (MHA) tools suite was chosen because its size and complexity is representative of similar industry frameworks. For the animation controls, $c \in \mathbb{R}^{174}$ can be divided into 97 primary controls and 77 so-called "tweaker" controls. They are combined in various ways to create 814 pose-space deformation (PSD) controls. The MetaHuman animation rig controls 24,049

vertices via 870 joints, each with 9 degrees of freedom (3 translation, 3 rotation, 3 scaling). For the sake of implementation, we implement the rig to output the 7,830 degrees of freedom associated with the joints instead of the 72,147 degrees of freedom associated with the vertices. This optimization ignores the fact that the PSD controls can also affect blendshape correctives; instead, it assumes that the final mesh can be obtained merely by skinning the joints. The discussions throughout this paper are valid in either case, but this simplification makes the optimization more tractable. The animation rig determines the joint degrees of freedom by multiplying the joint matrix by the PSD controls. The size $7{,}830 \times 814$ joint matrix has only 745,284 nonzero entries, meaning that $\theta_{\mathcal{R}} \in \mathbb{R}^{745284}$.

In order to differentiate the tracker via Equation 36b, $\frac{\partial \mathcal{R}}{\partial c} \in \mathbb{R}^{(7830,174)}$ and $\frac{\partial \mathcal{R}}{\partial \theta} \in \mathbb{R}^{(7830,745284)}$ would need to be computed. In order to make $\frac{\partial \mathcal{R}}{\partial \theta}$ tractible, each term in the sum in Equation 35 is minimized over only the nonzero θ in the columns of the joint matrix corresponding to the PSD controls that are not identically zero according to c^+. This greatly reduces the dimension of θ for any reasonable (c^+, v^+) pair. For example, the nineteen expressions (ignoring the neutral) used for the Simon-Says capture have their dimensionality reduced to a far more tractible 1589, 6456, 1470, 1470, 9470, 15252, 11465, 7890, 7939, 2868, 4050, 3198, 12936, 4051, 7262, 3758, 7506, 2594, 2594, respectively. Of course, reducing each expression to a few thousand parameters means that hundreds of expressions (depending on expression overlap) would be needed to cover the full rig. We circumvent this by limiting our expression set to cover only the most important controls, noting that this leaves $\theta_{\mathcal{T}}$ fixed to its $\theta_{\mathcal{R}}$ values for the parameters that do not appear in the expression set (in the spirit of the last term of Equation 35).

5.2. Tracker variants

The following is a detailed description of the variations of Equation 35 we use for our fine-tuning strategy. For each expression, i.e. each term in the sum in Equation 35, let $\mathcal{H}$ be a diagonal matrix of Heaviside functions $H(|c_i^+|)$. Let $\mathcal{T}_{\mathcal{H}} = \mathcal{H}\mathcal{T}$ represent the filtering of the tracker to zero out entries that are zero in c^+. Let $\mathcal{H}_D$ be the decimation of $\mathcal{H}$ into a wide matrix via the elimination of rows that are entirely full of zeros; then, $c_D = \mathcal{H}_D c$ is the subset of c containing all relevant controls, and $\mathcal{H}_D\mathcal{T}$ is a similarly decimated tracker. Although $\mathcal{T}_{\mathcal{H}}$ and $\mathcal{H}_D\mathcal{T}$ are essentially equivalent, $\mathcal{H}_D\mathcal{T}$ facilitates code optimizations. Let θ_D be the subset of the rig parameters that depends on c_D, let $\mathcal{R}_D(c_D;\theta_D)$ be the subset of the rig that can be modified by c_D, and let $\mathcal{T}_D(I;\theta_D)$ represent a reduced tracker that only considers the reduced degrees of freedom from c_D. In contrast to $\mathcal{T}_{\mathcal{H}}$, $\mathcal{T}_D$ is not allowed to use filtered out degrees of freedom in order to explain the geometry.

The γ_1 term in Equation 35 can use any of the three trackers, decimating c^+ for $\mathcal{T}_D$. The γ_2 term in Equation 35 can use any of the three trackers modifying $\mathcal{R}$ to $\mathcal{R}_D$ for $\mathcal{T}_D$; in addition, v^+ should be decimated if the output of $\mathcal{R}_D$ is a decimated version of the output of $\mathcal{R}$. Superscripts on γ (i.e. γ, $\gamma^{\mathcal{H}}$, γ^D) will be used to indicate which tracker was used in a specific term in Equation 35. Equation 36 can be used for any of the three trackers, modifying $\mathcal{R}$ to $\mathcal{R}_D$ for $\mathcal{T}_D$; in addition, v^+ should be decimated if the output of $\mathcal{R}_D$ is. Inserting $\mathcal{T}_{\mathcal{H}}$ into Equation 36b and using the normal equations on $\frac{\partial \mathcal{R}}{\partial c}\mathcal{H}$ leads to a coefficient matrix with nonzero entries corresponding to the normal equations for $\mathcal{R}_D$, i.e. $\frac{\partial \mathcal{T}_{\mathcal{H}}}{\partial \theta}$ agrees with $\frac{\partial \mathcal{T}_D}{\partial \theta}$ when it is nonzero. This highlights the aforementioned code optimizations for $\frac{\partial \mathcal{T}_{\mathcal{H}}}{\partial \theta}$, which consist of replacing $\mathcal{R}$ with $\mathcal{R}_D$ and $\mathcal{T}_{\mathcal{H}}$ with $\mathcal{H}_D\mathcal{T}$.

5.3. Threshold Conditions

Perhaps one of the most important things to keep in mind when perturbing $\theta_A \to \hat{\theta}_A$ or $\theta_S \to \hat{\theta}_S$ without the ability to re-optimize ψ via

$$\min_{\psi} \sum_k ||\mathcal{T}(I_k;\psi) - c_k||_2^2 \tag{37}$$

or even to understand how ψ was optimized is that sensitive threshold conditions may have been used. For example, a lip sealing constraint may activate when the lips are close enough together but otherwise do nothing. This means that a small perturbation of the rig could cause lips that were previously sealed via this constraint to instead be noticeably open even though their actual position prior to the constraint activation only changes by a small amount.

5.4. Closed-Source Tracker Geometry Reconstruction

Information about the geometry reconstruction performance of perturbed closed-source trackers can be found in Figures 6 and 7.

5.5. Experimental setup

For running the tracker refinement process, we use the SGD optimizer PyTorch implementation for with $\texttt{lr} = 1\text{E}1$ for the MetaHuman rig framework and $\texttt{lr} = 1\text{E}{-2}$ for the game character rig framework. For the MetaHuman rig framework, we use $\gamma_1^D = 1\text{E}4, \gamma_2^D = 1\text{E}0, \gamma_\epsilon = 1\text{E}0, \texttt{iters} = 50$ for Stage 1, $\gamma_1^{\mathcal{H}} = 1\text{E}3, \gamma_2^{\mathcal{H}} = 1\text{E}{-1}, \gamma_\epsilon = 1\text{E}{-1}, \texttt{iters} = 50$ for Stages 2 and 3, and we iterate over 30 different spurious controls with $\gamma_1^{\mathcal{H}} = 1\text{E}3, \gamma_2^{\mathcal{H}} = 1\text{E}{-1}, \gamma_\epsilon = 1\text{E}{-1}, \texttt{iters} = 10$ for stage 4. To choose spurious controls for Stage 4, we take the most inaccurate control over all calibration expressions compared against the reference controls c_k. For the game character rig framework, we use $\gamma_1^D = 1\text{E}4, \gamma_2^D = 1\text{E}0, \gamma_\epsilon = 1\text{E}{-2}, \texttt{iters} = 50$ for Stage 1, $\gamma_1^{\mathcal{H}} = 1\text{E}3, \gamma_2^{\mathcal{H}} = 1\text{E}{-1}, \gamma_\epsilon = 1\text{E}{-3}, \texttt{iters} = 50$ for Stages 2 and 3, and we iterate over 30 different spurious controls with $\gamma_1^{\mathcal{H}} = 1\text{E}3, \gamma_2^{\mathcal{H}} = 1\text{E}{-1}, \gamma_\epsilon = 1\text{E}{-3}, \texttt{iters} = 20$ for stage 4.

Our open-source tracker we use in our experiments runs the L-BFGS optimizer with $\texttt{tolerance\_grad} = 1\text{E}{-8}, \texttt{max\_iter} = 50$ to optimize rig controls to match some geometry. We assume all geometry is in the canonical rigid frame of the rig. We use a mean squared L2 loss on the per-vertex error, and L1 regularization on the controls with a penalty weight of $1\text{E}{-4}$ to encourage sparsity in solved controls. Occasionally, the tracker will fail to converge to an acceptable solution for some frame of a performance. If this occurs, we average the results from the surrounding frames if a restart did not help.

Figure 6: *A 573 frame pangram was used to test the tracker's ability to invert the rig and match the reconstructed geometry. The tracker was mostly able to adequately minimize geometry errors using any of θ_A, $\hat{\theta}_A$, θ_S, $\hat{\theta}_S$, even though the output controls (and thus the semantic interpretation) can vary significantly. See also Figure 7.*

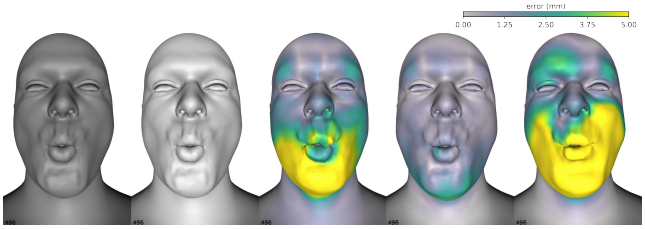

Figure 7: *Ground-truth reconstructed geometry (left) as compared to the geometry output by the tracker using θ_A, $\hat{\theta}_A$, θ_S, $\hat{\theta}_S$, respectively. This corresponds to frame 100, which was chosen because it has relatively large errors, of Figure 6. Colored regions indicate reconstruction error according to the colorbar.*

6. Experiments: Additional Details

6.1. Open-source Tracker

Figures 8, 9, 10, and 11 show the step-by-step improvement in semantic accuracy from stages 1, 2, 3, and 4 (respectively) of our proposed optimization strategy. These plots demonstrate that each stage of our method successfully improves upon the best result from the previous stage.

6.2. Closed-source Tracker

Figures 12, 13, and 14 show the step-by-step improvement in semantic accuracy from stages 2, 3, and 4 (respectively) of our proposed optimization strategy. Stage 1 is not shown because it is not applied to closed-source trackers. These plots demonstrate that each stage of our method successfully improves upon the best result from the previous stage. Additionally, a summary of how the full tracker T improves over all stages is shown in Figure 15.

6.3. Semantic FLAME rig

We explain details of the creation of our semantic FLAME rig. [LBB*17] notes that a more semantic controls space for this rig framework might be preferred for animation. We choose a reference animation rig which has the semantic controls space we want to emulate. We first fit the FLAME rigid and identity parameters to

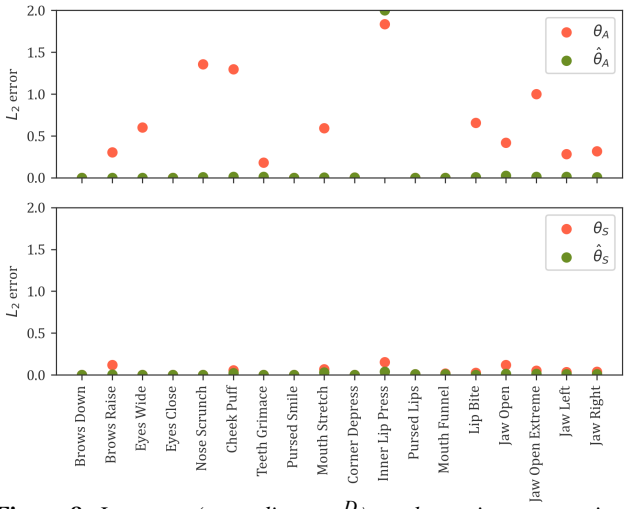

Figure 8: *L_2 errors (according to γ_1^D) on the various expressions before (red) and after (green) optimizing $\theta_A \to \hat{\theta}_A$ (top) and $\theta_S \to \hat{\theta}_S$ (bottom) using only the primary controls on each expression via γ_1^D and γ_2^D.*

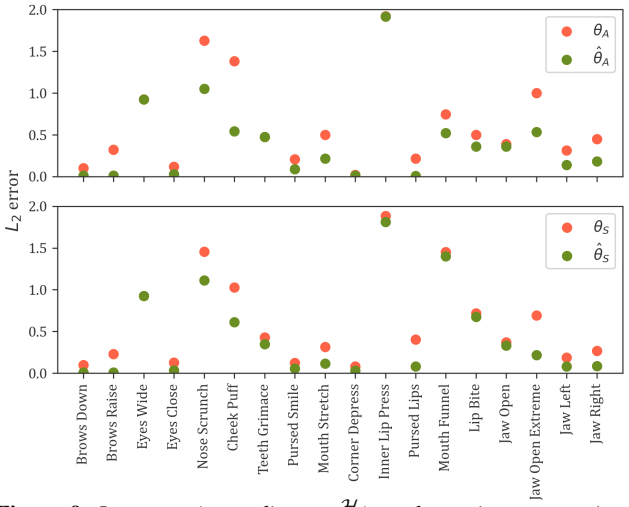

Figure 9: *L_2 errors (according to $\gamma_1^{\mathcal{H}}$) on the various expressions before (red) and after (green) optimizing $\theta_A \to \hat{\theta}_A$ (top) and $\theta_S \to \hat{\theta}_S$ (bottom) using the full rig while filtering the primary controls on each expression via $\gamma_1^{\mathcal{H}}$ and $\gamma_2^{\mathcal{H}}$.*

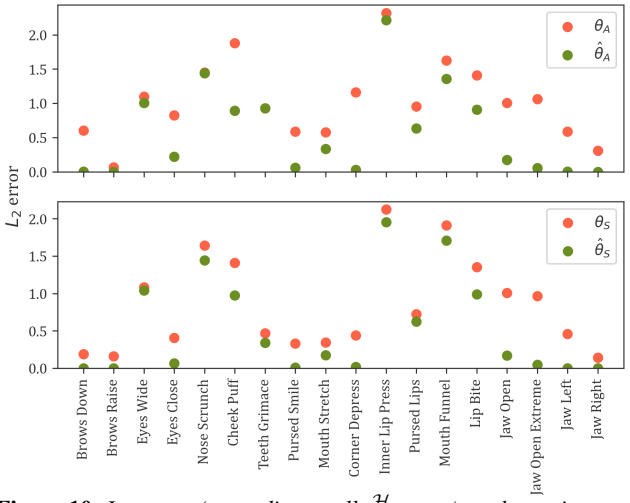

Figure 10: *L_2 errors (according to all $\gamma_1^{\mathcal{H}}$ terms) on the various expressions before (red) and after (green) optimizing $\theta_A \to \hat{\theta}_A$ (top) and $\theta_S \to \hat{\theta}_S$ (bottom) using the full rig while filtering the primary controls on each expression via $\gamma_1^{\mathcal{H}}$ and $\gamma_2^{\mathcal{H}}$ and additionally filtering various spurious controls via an additional $\gamma_1^{\mathcal{H}}$ term. Note that the minimization is still only considering the columns corresponding to the primary controls.*

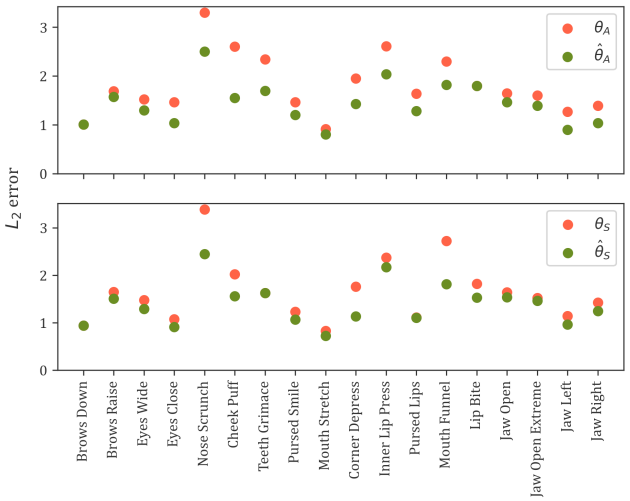

Figure 11: *L_2 errors (according to all γ_1 terms) on the various expressions before (red) and after (green) optimizing $\theta_A \to \hat{\theta}_A$ (top) and $\theta_S \to \hat{\theta}_S$ (bottom) using the full rig while filtering the primary controls on each expression via $\gamma_1^{\mathcal{H}}$ and $\gamma_2^{\mathcal{H}}$ and additionally filtering various spurious controls via an additional $\gamma_1^{\mathcal{H}}$ term. The difference between this figure and Figure 10 is that the minimization is now considering the columns of the spurious controls while freezing the columns of the primary controls.*

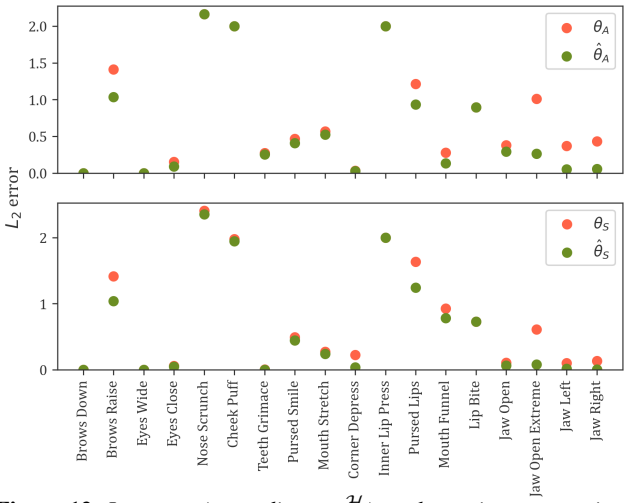

Figure 12: *L_2 errors (according to $\gamma_1^{\mathcal{H}}$) on the various expressions before (red) and after (green) optimizing $\theta_A \to \hat{\theta}_A$ (top) and $\theta_S \to \hat{\theta}_S$ (bottom) using the full rig while filtering the primary controls on each expression via $\gamma_1^{\mathcal{H}}$ and $\gamma_2^{\mathcal{H}}$.*

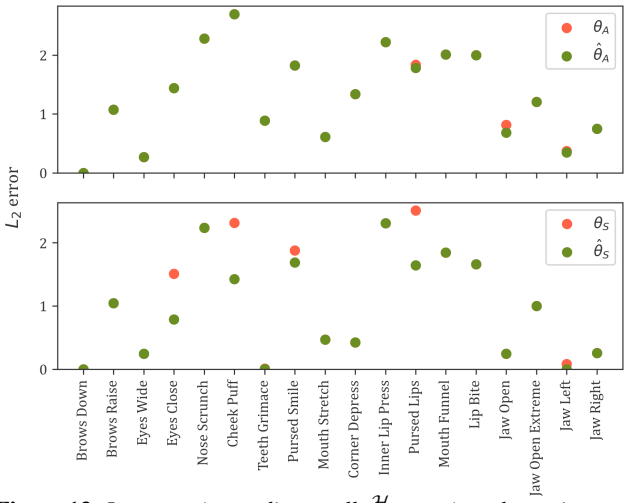

Figure 13: *L_2 errors (according to all $\gamma_1^{\mathcal{H}}$ terms) on the various expressions before (red) and after (green) optimizing $\theta_A \to \hat{\theta}_A$ (top) and $\theta_S \to \hat{\theta}_S$ (bottom) using the full rig while filtering the primary controls on each expression via $\gamma_1^{\mathcal{H}}$ and $\gamma_2^{\mathcal{H}}$ and additionally filtering various spurious controls via an additional $\gamma_1^{\mathcal{H}}$ term. Note that the minimization is still only considering the columns corresponding to the primary controls.*

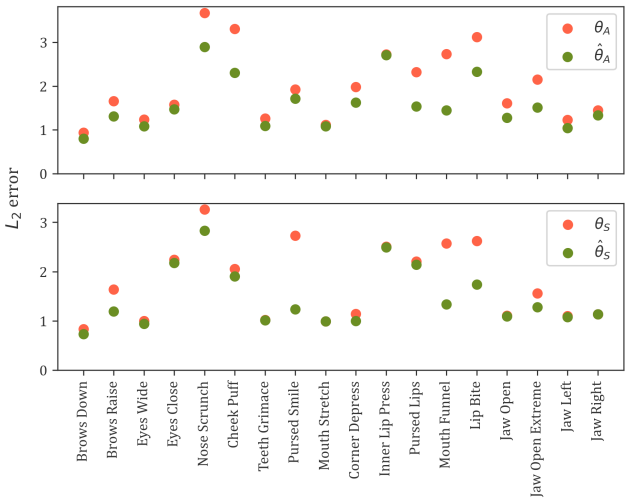

Figure 14: *L_2 errors (according to all γ_1 terms) on the various expressions before (red) and after (green) optimizing $\theta_A \rightarrow \hat{\theta}_A$ (top) and $\theta_S \rightarrow \hat{\theta}_S$ (bottom) using the full rig while filtering the primary controls on each expression via $\gamma_1^{\mathcal{H}}$ and $\gamma_2^{\mathcal{H}}$ and additionally filtering various spurious controls via an additional $\gamma_1^{\mathcal{H}}$ term. The difference between this figure and Figure 13 is that the minimization is now considering the columns of the spurious controls while freezing the columns of the primary controls.*

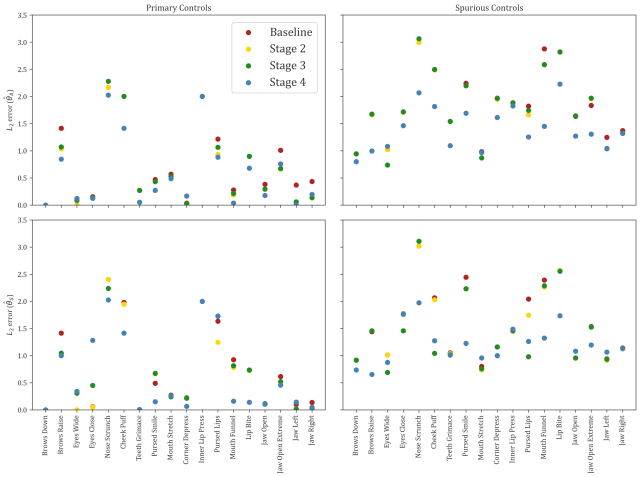

Figure 15: *Summary of the improvement in the L_2 errors (according to γ_1) throughout all three stages (there is no first stage on a closed-source tracker) of the process (see Figures 12 to 14). The improvement in the primary controls is shown to the left, and the improvement in the spurious controls is shown to the right. Optimizing $\theta_A \rightarrow \hat{\theta}_A$ is shown on the top, and optimizing $\theta_S \rightarrow \hat{\theta}_S$ is shown on the bottom.*

the neutral of our chosen animation rig. Then, for a subset of relevant controls we want to utilize, we register FLAME PCA coefficients to the deformed animation rig for that basis expression. This constitutes a linearized mapping from the semantic control space of the reference rig to the FLAME PCA space. To capture relationships between the controls or nonlinearities in the rig, a neutral network perturbation to this mapping can be learned given (controls, geometry) pairs from the reference animation rig, by fitting the FLAME coefficients to each of those expressions. Thus, the Semantic FLAME rig takes as input semantic control parameters, maps them to FLAME PCA parameters, and evaluates the FLAME model to get geometry in a fully differentiable manner.

6.4. Puppeteering a game character

For our puppeteering examples, we designed a set of basis expressions that better-covered the rig controls of the virtual character. In total, fifteen expressions were used. Eleven were expressions similar to those designed for the MetaHuman framework (“neutral”, “eyes close”, “eyes wide”, “brows raise”, “brows down”, “teeth grimace”, “pursed smile”, “corner depress”, “jaw open”, “jaw left”, “jaw right”), and four were phoneme-level expressions unique to the game rig “OO phoneme”, “CH phoneme”, “M/B/P phoneme”, “F/V phoneme”).

To create a game character representation for the performer, we first creating a correspondence between the game/VR character’s triangle mesh and the performer’s triangle mesh in order to assign a game/VR character triangle mesh to the performer’s neutral geometry. Note that the teeth were ignored, since they were not used in the tracking of the performer. Then, the game/VR character rig was volumetrically morphed to the performer’s new game/VR character consistent neutral geometry.

6.5. Animation-focused Quantitative Metrics

What is classified as a “tweaker” control is up to some interpretation. For our definitions, we consulted with technical artists and determined a classification of primary versus tweaker controls for each of our rig frameworks, which are released along with the code. The sparsity threshold was set at 0.1 for our experiments.

6.6. User Study

We conduct a user study to whether our modified tracked animations look more semantically accurate to viewers. 17 participants were presented with a source video of a pangram or ROM, a virtual avatar in our semantic FLAME system driven by an uncalibrated tracker, and the same virtual avatar driven by a calibrated tracker, and asked which video more closely semantically matches the expressions in the source video. 82% of calibrated videos were selected as preferred over the baseline, affirming that our claim of improved animation quality is meaningful in the eye of those viewing the animation.

References

[ASY*24] ATHAR S., SAITO S., YANG Z., PIDHORSKYI S., CAO C.: Bridging the gap: Studio-like avatar creation from a monocular phone

capture. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XII* (Berlin, Heidelberg, 2024), Springer-Verlag, p. 72–88. doi:10.1007/978-3-031-73254-6_5. 1

[Aut25] AUTODESK, INC.: Maya, 2025. URL: https://www.autodesk.com/products/maya. 1

[BB14] BEELER T., BRADLEY D.: Rigid stabilization of facial expressions. *ACM Trans. Graph. 33*, 4 (jul 2014). 1

[BBC*24] BAERT K., BHARADWAJ S., CASTAN F., MAUJEAN B., CHRISTIE M., ABREVAYA V., BOUKHAYMA A.: Spark: Self-supervised personalized real-time monocular face capture. doi:10.1145/3680528.3687704. 1

[BCGF19] BAO M., CONG M., GRABLI S., FEDKIW R.: High-quality face capture using anatomical muscles. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 10794–10803. doi:10.1109/CVPR.2019.01106. 1

[Ble25] BLENDER ONLINE COMMUNITY: Blender, 2025. URL: https://www.blender.org/. 1

[BLW*24] BUEHLER M. C., LI G., WOOD E., HELMINGER L., CHEN X., SHAH T., WANG D., GARBIN S., ORTS-ESCOLANO S., HILLIGES O., LAGUN D., RIVIERE J., GOTARDO P., BEELER T., MEKA A., SARKAR K.: Cafca: High-quality novel view synthesis of expressive faces from casual few-shot captures. In *ACM SIGGRAPH Asia 2024 Conference Paper*. 2024. doi:10.1145/3680528.3687580. 1

[BODO20] BAILEY S. W., OMENS D., DILORENZO P., O'BRIEN J. F.: Fast and deep facial deformations. *ACM Transactions on Graphics 39*, 4 (Aug. 2020), 94:1–15. Presented at SIGGRAPH 2020, Washington D.C. URL: http://graphics.berkeley.edu/papers/Bailey-FDF-2020-07/, doi:10.1145/3386569.3392397. 1

[Bro65] BROYDEN C. G.: A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation 19* (1965), 577–593. 1, 2

[BSS*23] BUEHLER M. C., SARKAR K., SHAH T., LI G., WANG D., HELMINGER L., ORTS-ESCOLANO S., LAGUN D., HILLIGES O., BEELER T., MEKA A.: Preface: A data-driven volumetric prior for few-shot ultra high-resolution face synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023). 1

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3d faces. In *SIGGRAPH* (1999). 1

[BZH*23] BHARADWAJ S., ZHENG Y., HILLIGES O., BLACK M. J., ABREVAYA V. F.: Flare: Fast learning of animatable and relightable mesh avatars. *ACM Transactions on Graphics 42* (Dec. 2023), 15. doi:https://doi.org/10.1145/3618401. 1

[CBE*15] CONG M., BAO M., E J. L., BHAT K. S., FEDKIW R.: Fully automatic generation of anatomical face simulation models. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (New York, NY, USA, 2015), SCA '15, Association for Computing Machinery, p. 175–183. 1

[CEM*22] CHOI B., EOM H., MOUSCADET B., CULLINGFORD S., MA K., GASSEL S., KIM S., MOFFAT A., MAIER M., REVELANT M., LETTERI J., SINGH K.: Animatomy: an animator-centric, anatomically inspired system for 3d facial modeling, animation and transfer. In *SIGGRAPH Asia 2022 Conference Papers* (New York, NY, USA, 2022), SA '22, Association for Computing Machinery. 1

[CSK*22] CAO C., SIMON T., KIM J. K., SCHWARTZ G., ZOLLHOEFER M., SAITO S.-S., LOMBARDI S., WEI S.-E., BELKO D., YU S.-I., SHEIKH Y., SARAGIH J.: Authentic volumetric avatars from a phone scan. *ACM Trans. Graph. 41*, 4 (jul 2022). 1

[DAT*23] DIB A., AHN J., THÉBAULT C., GOSSELIN P.-H., CHEVALLIER L.: S2f2: Self-supervised high fidelity face reconstruction from monocular image. In *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG)* (2023), IEEE Press, p. 1–8. doi:10.1109/FG57933.2023.10042713. 1

[Dav91] DAVIDON W. C.: Variable metric method for minimization. *SIAM Journal on Optimization 1*, 1 (1991), 1–17. doi:10.1137/0801001. 2

[DBA*21] DIB A., BHARAJ G., AHN J., THÉBAULT C., GOSSELIN P., ROMEO M., CHEVALLIER L.: Practical face reconstruction via differentiable ray tracing. *Computer Graphics Forum 40*, 2 (2021), 153–164. doi:https://doi.org/10.1111/cgf.142622. 1

[DBB22] DANECEK R., BLACK M. J., BOLKART T.: EMOCA: Emotion driven monocular face capture and animation. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 20311–20322. 1

[DGHG*24] DIB A., GUSTAVO HAFEMANN L., GOT E., ANDERSON T., FADAEINEJAD A., CRUZ R. M. O., CARBONNEAU M.-A.: Mosar: Monocular semi-supervised model for avatar reconstruction using differentiable shading. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024), pp. 1770–1780. doi:10.1109/CVPR52733.2024.00174. 1

[DHT*00] DEBEVEC P., HAWKINS T., TCHOU C., DUIKER H.-P., SAROKIN W., SAGAR M.: Acquiring the reflectance field of a human face. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (USA, 2000), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., p. 145–156. 1

[DTA*21] DIB A., THEBAULT C., AHN J., GOSSELIN P.-H., THEOBALT C., CHEVALLIER L.: Towards High Fidelity Monocular Face Reconstruction with Rich Reflectance using Self-supervised Learning and Ray Tracing . In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (Los Alamitos, CA, USA, Oct. 2021), IEEE Computer Society, pp. 12799–12809. doi:10.1109/ICCV48922.2021.01258. 1

[DWS*23] DUAN H.-B., WANG M., SHI J.-C., CHEN X.-C., CAO Y.-P.: Bakedavatar: Baking neural fields for real-time head avatar synthesis. *ACM Trans. Graph. 42*, 6 (sep 2023). URL: https://doi.org/10.1145/3618399, doi:10.1145/3618399. 1

[DYX*19] DENG Y., YANG J., XU S., CHEN D., JIA Y., TONG X.: Accurate 3D Face Reconstruction With Weakly-Supervised Learning: From Single Image to Image Set . In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (Los Alamitos, CA, USA, June 2019), IEEE Computer Society, pp. 285–295. doi:10.1109/CVPRW.2019.00038. 1

[Epi25] EPIC GAMES: Metahuman animator, 2025. URL: https://www.unrealengine.com/en-US/digital-humans. 1

[EST*20] EGGER B., SMITH W. A. P., TEWARI A., WUHRER S., ZOLLHOEFER M., BEELER T., BERNARD F., BOLKART T., KORTYLEWSKI A., ROMDHANI S., THEOBALT C., BLANZ V., VETTER T.: 3d morphable face models—past, present, and future. *ACM Trans. Graph. 39*, 5 (jun 2020). 1

[Fle87] FLETCHER R.: *Practical Methods of Optimization*. John Wiley and Sons, 1987. 2

[FP63] FLETCHER R., POWELL M. J. D.: A rapidly convergent descent method for minimization. *The Computer Journal 6*, 2 (08 1963), 163–168. doi:10.1093/comjnl/6.2.163. 2

[Gea71] GEAR C. W.: *Numerical initial value problems in ordinary differential equations*. Longman Higher Education, Harlow, England, 1971. 5

[GFT*11] GHOSH A., FYFFE G., TUNWATTANAPONG B., BUSCH J., YU X., DEBEVEC P.: Multiview face capture using polarized spherical gradient illumination. *ACM Trans. Graph. 30*, 6 (Dec. 2011), 1–10. doi:10.1145/2070781.2024163. 1

[GPL*21] GRASSAL P.-W., PRINZLER M., LEISTNER T., ROTHER C., NIESSNER M., THIES J.: Neural head avatars from monocular rgb videos. *arXiv preprint arXiv:2112.01554* (2021). 1

[GTZN21] GAFNI G., THIES J., ZOLLHÖFER M., NIESSNER M.: Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2021), pp. 8649–8658. 1

[GVWT13] GARRIDO P., VALGAERT L., WU C., THEOBALT C.: Reconstructing detailed dynamic face geometry from monocular video. *ACM Trans. Graph. 32*, 6 (nov 2013). 1

[HLX24] HAN Y., LYU J., XU F.: High-quality facial geometry and appearance capture at home. 1

[HSA*24] HEWITT C., SALEH F., ALIAKBARIAN S., PETIKAM L., REZAEIFAR S., FLORENTIN L., HOSENIE Z., CASHMAN T. J., VALENTIN J., COSKER D., BALTRUSAITIS T.: Look ma, no markers: holistic performance capture without the hassle. *ACM Trans. Graph. 43*, 6 (Nov. 2024). doi:10.1145/3687772. 1

[KDP*24] KAVAN L., DOUBLESTEIN J., PRAZAK M., CIOFFI M., ROBLE D.: Compressed skinning for facial blendshapes. In *ACM SIGGRAPH 2024 Conference Papers* (New York, NY, USA, 2024), SIGGRAPH '24, Association for Computing Machinery. doi:10.1145/3641519.3657477. 1

[KKLD23] KERBL B., KOPANAS G., LEIMKUEHLER T., DRETTAKIS G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph. 42*, 4 (July 2023). doi:10.1145/3592433. 1

[KQG*23] KIRSCHSTEIN T., QIAN S., GIEBENHAIN S., WALTER T., NIESSNER M.: Nersemble: Multi-view radiance field reconstruction of human heads. *ACM Trans. Graph. 42*, 4 (jul 2023). URL: https://doi.org/10.1145/3592455, doi:10.1145/3592455. 1

[LAGP09] LI H., ADAMS B., GUIBAS L. J., PAULY M.: Robust single-view geometry and motion reconstruction. *ACM Trans. Graph. 28*, 5 (dec 2009), 1–10. 1

[LBB*17] LI T., BOLKART T., BLACK M. J., LI H., ROMERO J.: Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia) 36*, 6 (2017), 194:1–194:17. URL: https://doi.org/10.1145/3130800.3130813. 1, 8

[LKA*17] LAINE S., KARRAS T., AILA T., HERVA A., SAITO S., YU R., LI H., LEHTINEN J.: Production-level facial performance capture using deep convolutional neural networks. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (New York, NY, USA, 2017), SCA '17, Association for Computing Machinery. 1

[LN89] LIU D. C., NOCEDAL J.: On the limited memory bfgs method for large scale optimization. *Mathematical Programming 45*, 1 (08 1989), 503–528. doi:10.1007/BF01589116. 2, 6

[LRF*23] LEI B., REN J., FENG M., CUI M., XIE X.: A Hierarchical Representation Network for Accurate and Detailed Face Reconstruction from In-The-Wild Images . In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Los Alamitos, CA, USA, June 2023), IEEE Computer Society, pp. 394–403. doi:10.1109/CVPR52729.2023.00046. 1

[LSSS18] LOMBARDI S., SARAGIH J., SIMON T., SHEIKH Y.: Deep appearance models for face rendering. *ACM Trans. Graph. 37*, 4 (jul 2018). 1

[LWP10] LI H., WEISE T., PAULY M.: Example-based facial rigging. *ACM Trans. Graph. 29*, 4 (jul 2010). 1

[MLL*24] MING X., LI J., LING J., ZHANG L., XU F.: High-quality mesh blendshape generation from face videos via neural inverse rendering. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXX* (Berlin, Heidelberg, 2024), Springer-Verlag, p. 106–125. doi:10.1007/978-3-031-72897-6_7. 1

[MSS*21] MA S., SIMON T., SARAGIH J., WANG D., LI Y., TORRE F. L., SHEIKH Y.: Pixel codec avatars. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Los Alamitos, CA, USA, jun 2021), IEEE Computer Society, pp. 64–73. 1

[MST*21] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM 65*, 1 (Dec. 2021), 99–106. doi:10.1145/3503250. 1

[NW06] NOCEDAL J., WRIGHT S. J.: *Numerical optimization*. Springer, 2006. 2

[OBP*12] ORVALHO V., BASTOS P., PARKE F., OLIVEIRA B., ALVAREZ X.: A Facial Rigging Survey. In *Eurographics 2012 - State of the Art Reports* (2012), Cani M.-P., Ganovelli F., (Eds.), The Eurographics Association. doi:/10.2312/conf/EG2012/stars/183-204. 1

[RFD*24] RETSINAS G., FILNTISIS P. P., DANECEK R., ABREVAYA V. F., ROUSSOS A., BOLKART T., MARAGOS P.: 3d facial expressions through analysis-by-neural-synthesis. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2024). 1

[SBL*23] SARKAR K., BÜHLER M. C., LI G., WANG D., VICINI D., RIVIERE J., ZHANG Y., ORTS-ESCOLANO S., GOTARDO P., BEELER T., MEKA A.: Litnerf: Intrinsic radiance decomposition for high-quality view synthesis and relighting of faces. In *SIGGRAPH Asia 2023 Conference Papers* (New York, NY, USA, 2023), SA '23, Association for Computing Machinery. doi:10.1145/3610548.3618210. 1

[SSS*24] SAITO S., SCHWARTZ G., SIMON T., LI J., NAM G.: Relightable gaussian codec avatars. In *CVPR* (2024). 1

[SWH*17] SAITO S., WEI L., HU L., NAGANO K., LI H.: Photorealistic facial texture inference using deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 2326–2335. doi:10.1109/CVPR.2017.250. 1

[TCS*23] TREVITHICK A., CHAN M., STENGEL M., CHAN E., LIU C., YU Z., KHAMIS S., CHANDRAKER M., RAMAMOORTHI R., NAGANO K.: Real-time radiance fields for single-image portrait view synthesis. *ACM Trans. Graph. 42*, 4 (July 2023). doi:10.1145/3592460. 1

[TRP*24] TEOTIA K., R M. B., PAN X., KIM H., GARRIDO P., ELGHARIB M., THEOBALT C.: Hq3davatar: High-quality implicit 3d head avatar. *ACM Trans. Graph. 43*, 3 (apr 2024). 1

[TRT*24] TAUBNER F., RAINA P., TULI M., TEH E. W., LEE C., HUANG J.: 3d face tracking from 2d video through iterative dense uv to image flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2024), pp. 1227–1237. 1

[TZK*17] TEWARI A., ZOLLHÖFER M., KIM H., GARRIDO P., BERNARD F., PÉREZ P., THEOBALT C.: Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 3735–3744. doi:10.1109/ICCV.2017.401. 1

[WBGB16] WU C., BRADLEY D., GROSS M., BEELER T.: An anatomically-constrained local deformation model for monocular face capture. *ACM Trans. Graph. 35*, 4 (jul 2016). 1

[WBH*22] WOOD E., BALTRUŠAITIS T., HEWITT C., JOHNSON M., SHEN J., MILOSAVLJEVIĆ N., WILDE D., GARBIN S., SHARP T., STOJILJKOVIĆ I., ET AL.: 3d face reconstruction with dense landmarks. In *European Conference on Computer Vision* (2022), Springer, pp. 160–177. 1

[WYL*23] WU S., YAN Y., LI Y., CHENG Y., ZHU W., GAO K., LI X., ZHAI G.: Ganhead: Towards generative animatable neural head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 437–447. 1

[WZZ*24] WANG Z., ZHU X., ZHANG T., WANG B., LEI Z.: 3d face reconstruction with the geometric guidance of facial part segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 1672–1682. 1

[XCL*24] XU Y., CHEN B., LI Z., ZHANG H., WANG L., ZHENG Z., LIU Y.: Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024). 1

[XGGZ24] XIANG J., GAO X., GUO Y., ZHANG J.: Flashavatar: High-fidelity head avatar with efficient gaussian embedding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2024). 1

[YZC*23] YANG L., ZOSS G., CHANDRAN P., GOTARDO P., GROSS M., SOLENTHALER B., SIFAKIS E., BRADLEY D.: An implicit physical face model driven by expression and style. In *SIGGRAPH Asia 2023 Conference Papers* (New York, NY, USA, 2023), SA '23, Association for Computing Machinery. 1

[YZF*23] YANG H., ZHENG M., FENG W., HUANG H., LAI Y.-K., WAN P., WANG Z., MA C.: Towards practical capture of high-fidelity relightable avatars. In *SIGGRAPH Asia 2023 Conference Proceedings* (2023). 1

[YZM*24] YANG H., ZHENG M., MA C., LAI Y.-K., WAN P., HUANG H.: Vrmm: A volumetric relightable morphable head model. In *ACM SIGGRAPH 2024 Conference Papers* (New York, NY, USA, 2024), SIGGRAPH '24, Association for Computing Machinery. doi:10.1145/3641519.3657406. 1

[ZAB*22] ZHENG Y., ABREVAYA V. F., BÜHLER M. C., CHEN X., BLACK M. J., HILLIGES O.: I M Avatar: Implicit morphable head avatars from videos. In *Computer Vision and Pattern Recognition (CVPR)* (2022). 1

[ZBT22a] ZIELONKA W., BOLKART T., THIES J.: Instant volumetric head avatars. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 4574–4584. URL: https://api.semanticscholar.org/CorpusID:253761096. 1

[ZBT22b] ZIELONKA W., BOLKART T., THIES J.: Towards metrical reconstruction of human faces. In *European Conference on Computer Vision* (2022). 1

[ZCL*23] ZHANG T., CHU X., LIU Y., LIN L., YANG Z., XU Z., CAO C., YU F., ZHOU C., YUAN C., LI Y.: Accurate 3d face reconstruction with facial component tokens. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (Los Alamitos, CA, USA, oct 2023), IEEE Computer Society, pp. 8999–9008. 1

[ZHXQ24] ZHOU M., HYDER R., XUAN Z., QI G.: Ultravatar: A realistic animatable 3d avatar diffusion model with authenticity guided textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2024), pp. 1238–1248. 1

[ZSCS04] ZHANG L., SNAVELY N., CURLESS B., SEITZ S. M.: Spacetime faces: high resolution capture for modeling and animation. *ACM Trans. Graph. 23*, 3 (aug 2004), 548–558. 1

[ZYW*23] ZHENG Y., YIFAN W., WETZSTEIN G., BLACK M. J., HILLIGES O.: Pointavatar: Deformable point-based head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023). 1