Face2Face, MoFA

Reading Group 2021-06-15

Benjamin Bray & Badrinath Singhal

(note: most of the text/images in this presentation were copied directly from one of the references!)

Primary References

- **[Thies 2016] "Face2Face: Real-time Face Capture and Reenactment of RGB Videos"**
- **[Tewari 2017] "MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction"**

Monocular Reconstruction

- [Garrido 2013] "Reconstructing Detailed Dynamic Face Geometry from Monocular Video"

Morphable Models

- [Li 2017] "FLAME"
- [Egger 2016] "Copula Eigenfaces with Attributes"
- [Egger 2019] "3D Morphable Models -- Past, Present, and Future" (survey paper)
- [Blanz 1999] "A Morphable Model for the Synthesis of 3D Faces"

Deformation Transfer

- [Sumner 2004] "Deformation Transfer for Triangle Meshes"
- [Roberts 2020] "Deformation Transfer Survey" (cites other recent papers)

Illumination

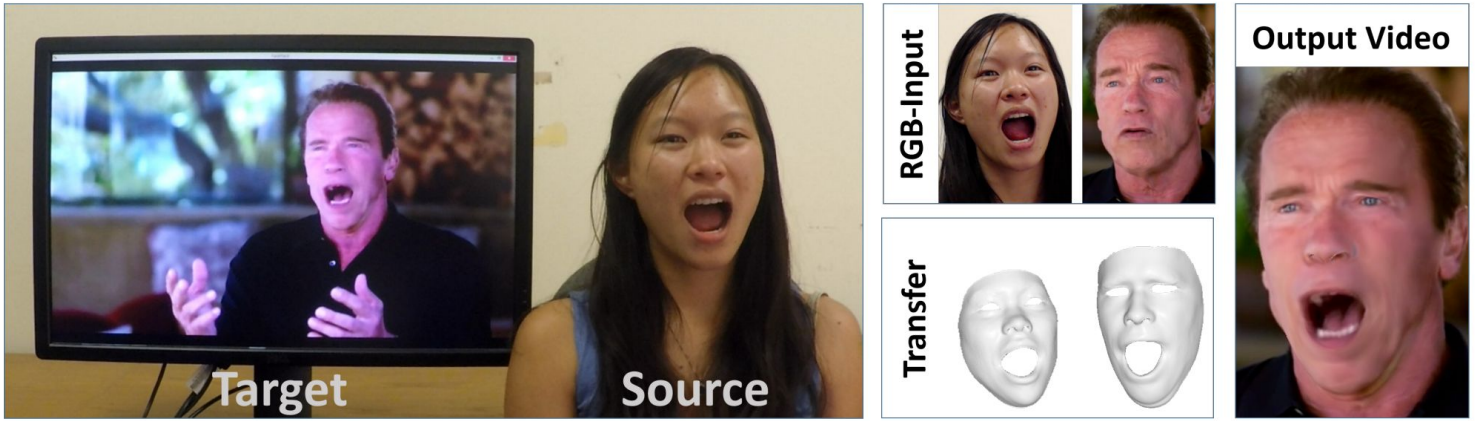
- [Green 2003] "Spherical Harmonic Lighting: The Gritty Details"

(if any of these topics interest you, we can read them in a future reading group!)

Face2Face

[Thies 2016] “Face2Face: Real-time Face Capture and Reenactment of RGB Videos”

Goal: Real-time Monocular Face Re-enactment



Proposed online reenactment setup: a monocular target video sequence (e.g., from Youtube) is reenacted based on the expressions of a source actor who is recorded live with a commodity webcam.

“Monocular” - Using only one camera to capture video

“Re-enactment” - Transfer facial expressions from the **source** to the **target**

- **Source** video is captured in real-time from a webcam
- **Target** video is recorded in advance (e.g. from YouTube)

(before Face2Face…)

Methods like [Garrido 2013] rely on complicated pre-processing steps.

- Complicated optimization procedure to perform deformation transfer
- Algorithm with several complicated steps to improve mesh quality
- Very slow, cannot run in real-time
- Captures very fine facial details (wrinkles, etc.)

Face2Face Algorithm

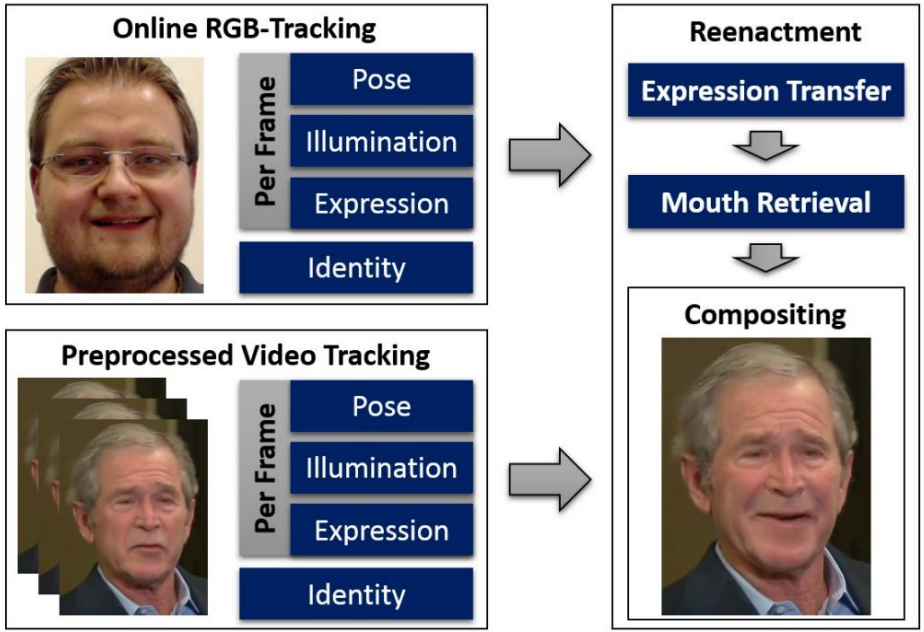
Preprocessing Target Video:

1. Identity estimation
2. “analysis-by-synthesis”
3. Isolate mouth frames

Real-Time:

4. Deformation Transfer
5. Mouth Interior Synthesis

(no deep learning involved!)

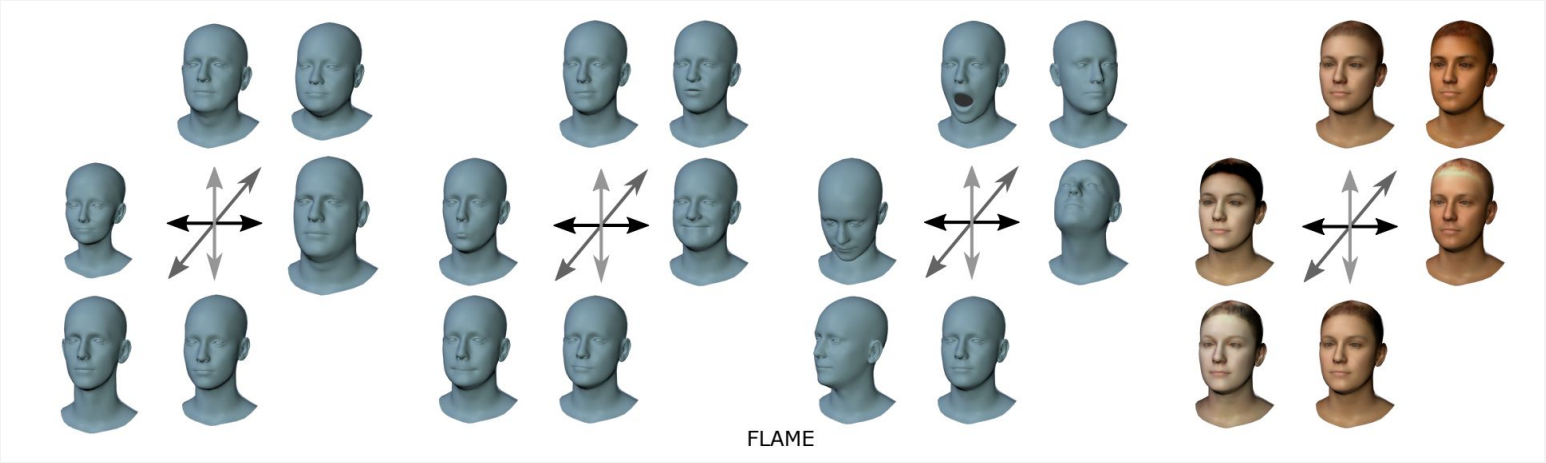


Background: 3D Morphable Models

Goal: Create new 3D face meshes from a set of **semantically meaningful parameters**.

- Option 1: 3D artist manually creates a parametric model using Maya, etc.
- Option 2: Automatically learn a 3D parametric model from a face dataset

Face2Face and MoFA can work with any parametric 3D morphable model.

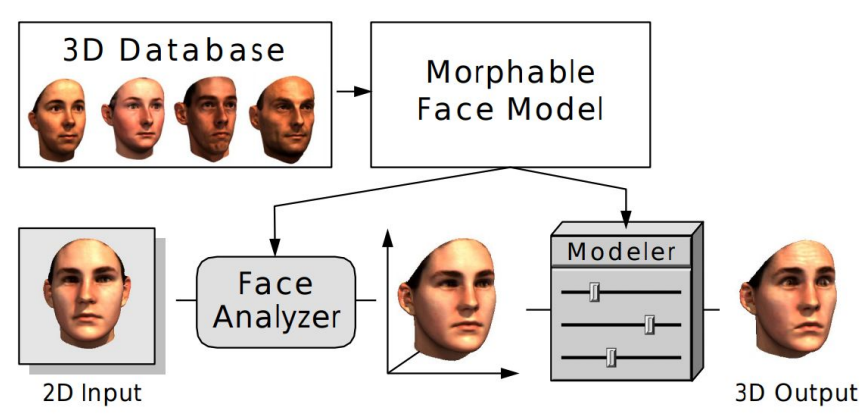


Background: 3D Morphable Models (Learn from Data)

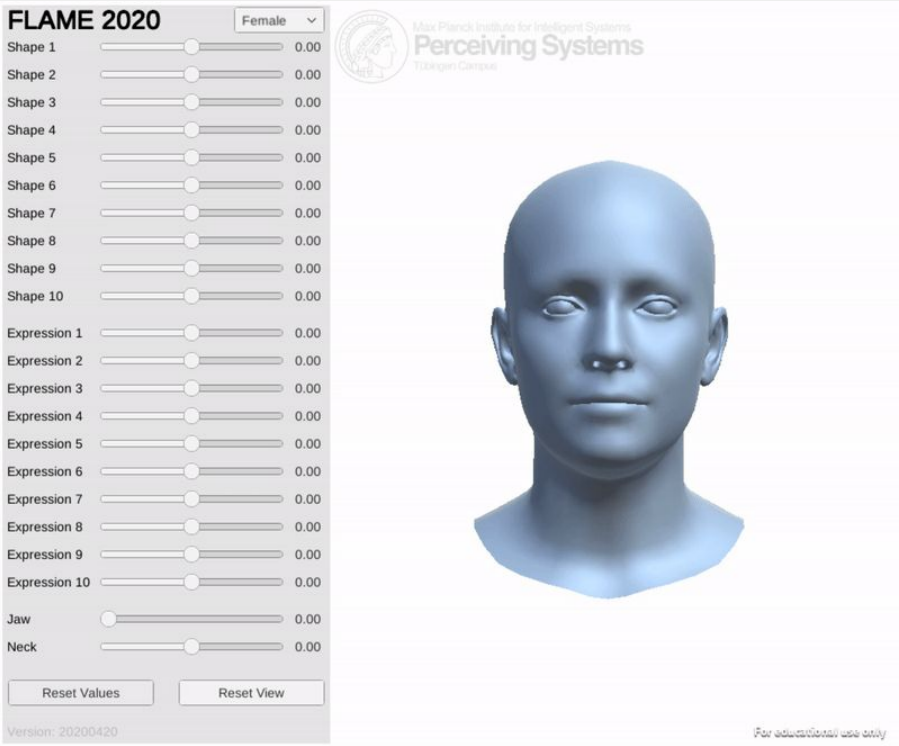
Input: Dataset of 3D Face Meshes

Output: Parametric Model for Generating new 3D Face Meshes

Algorithms: PCA, Autoencoders

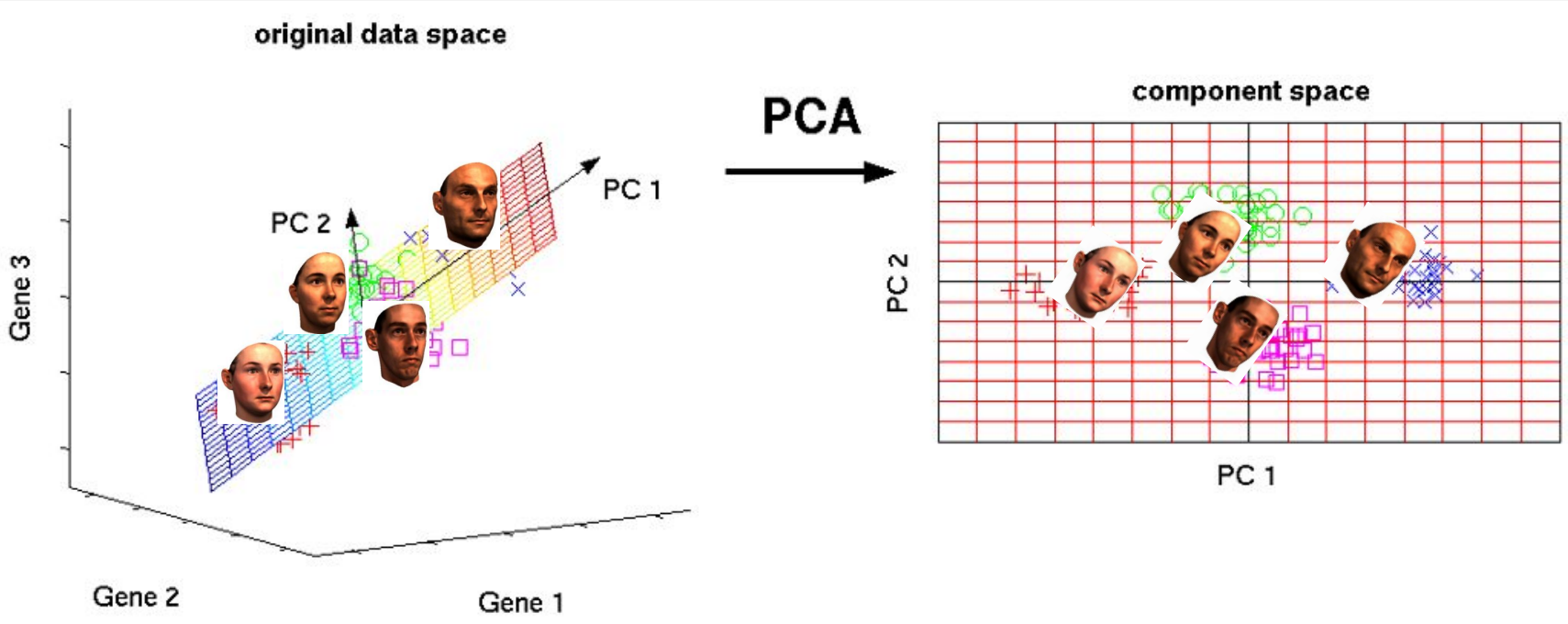


[Blanz 1999]



[Li 2017]

Background: PCA / Eigenfaces



https://www.kaggle.com/parulpandey/part1-visualizing-kannada-mnist-with-pca
https://setosa.io/ev/principal-component-analysis

Background: PCA / Eigenfaces

Eigenfaces in 2D (https://pydeep.readthedocs.io/en/latest/tutorials/PCA_eigenfaces.html)

Background: PCA

Theorem 12.2.1. *Suppose we want to find an orthogonal set of L linear basis vectors $\mathbf{w}_j \in \mathbb{R}^D$, and the corresponding scores $\mathbf{z}_i \in \mathbb{R}^L$, such that we minimize the average* **reconstruction error**

$$J(\mathbf{W}, \mathbf{Z}) = \frac{1}{N} \sum_{i} ||\mathbf{x}_i - \hat{\mathbf{x}}_i||^2 \tag{12.26}$$

where $\hat{\mathbf{x}}_i = \mathbf{W}\mathbf{z}_i$, subject to the constraint that $\mathbf{W}$ is orthonormal. Equivalently, we can write this objective as follows:

$$J(\mathbf{W}, \mathbf{Z}) = ||\mathbf{X} - \mathbf{W}\mathbf{Z}^T||_F^2 \tag{12.27}$$

The optimal solution is obtained by setting $\hat{\mathbf{W}} = \mathbf{V}_L$, where $\mathbf{V}_L$ contains the L eigenvectors with largest eigenvalues of the empirical covariance matrix, $\hat{\mathbf{\Sigma}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T$. (We assume the $\mathbf{x}_i$ have zero mean, for notational simplicity.) Furthermore, the optimal low-dimensional encoding of the data is given by $\hat{\mathbf{z}}_i = \mathbf{W}^T \mathbf{x}_i$, which is an orthogonal projection of the data onto the column space spanned by the eigenvectors.

Background: Generative Models (*"analysis-by-synthesis"*)

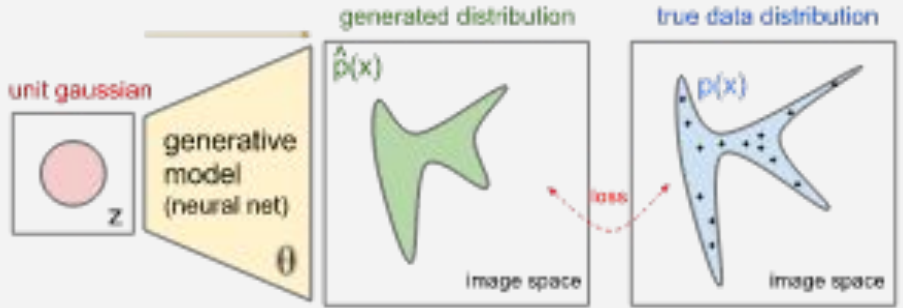
Fit a parametric face model to image and video data by optimizing the alignment between the projected model and the image

- Assume training data X comes from a probability distribution P(X ; θ)
  - parameters θ determine the shape of the distribution
- Normally, assume every X has a **latent parameter vector Z**

$$P(X ; \theta) = \int P(X,Z ; \theta)\, dZ = \int P(X \mid Z; \theta)\, P(Z) = f(\theta)$$

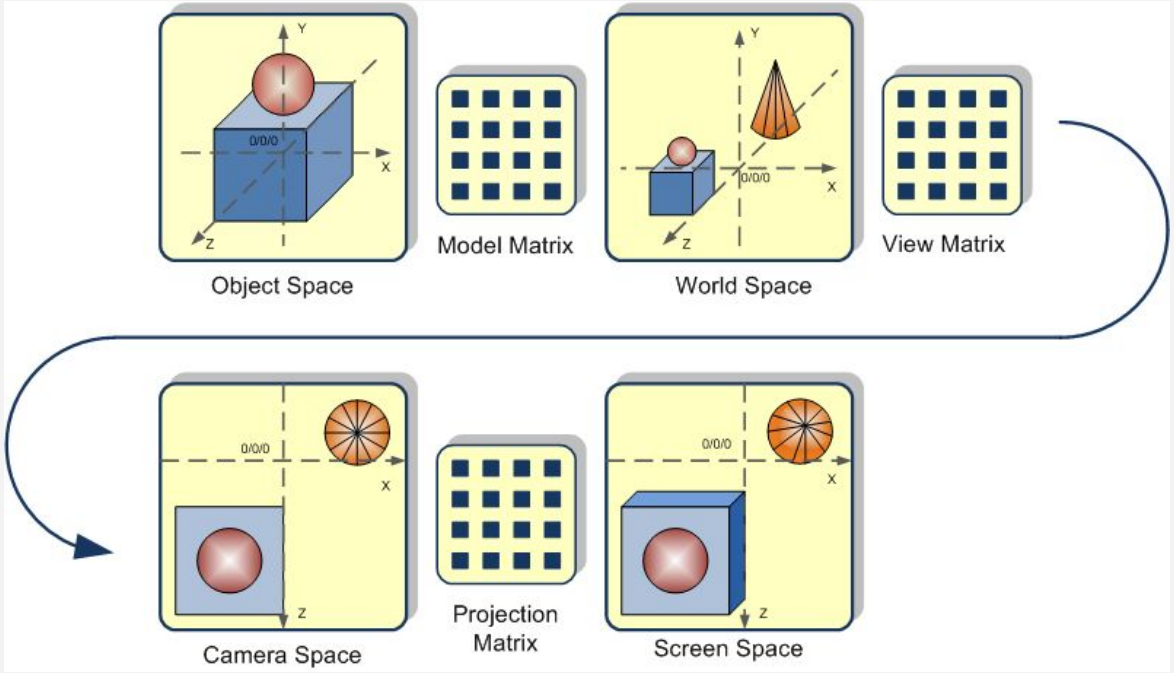
  - P(Z) is a standard multivariate Gaussian distribution
  - P(X | Z ; θ) a **deterministic, generative model** (usually a neural network)
- Find the **Maximum Likelihood Estimate (MLE)** by solving an **optimization problem**
  - Global θ for the entire dataset
  - Latent Z for each data point X

$$\theta_{ML} = \arg\max_{\theta} P(X|\theta)$$



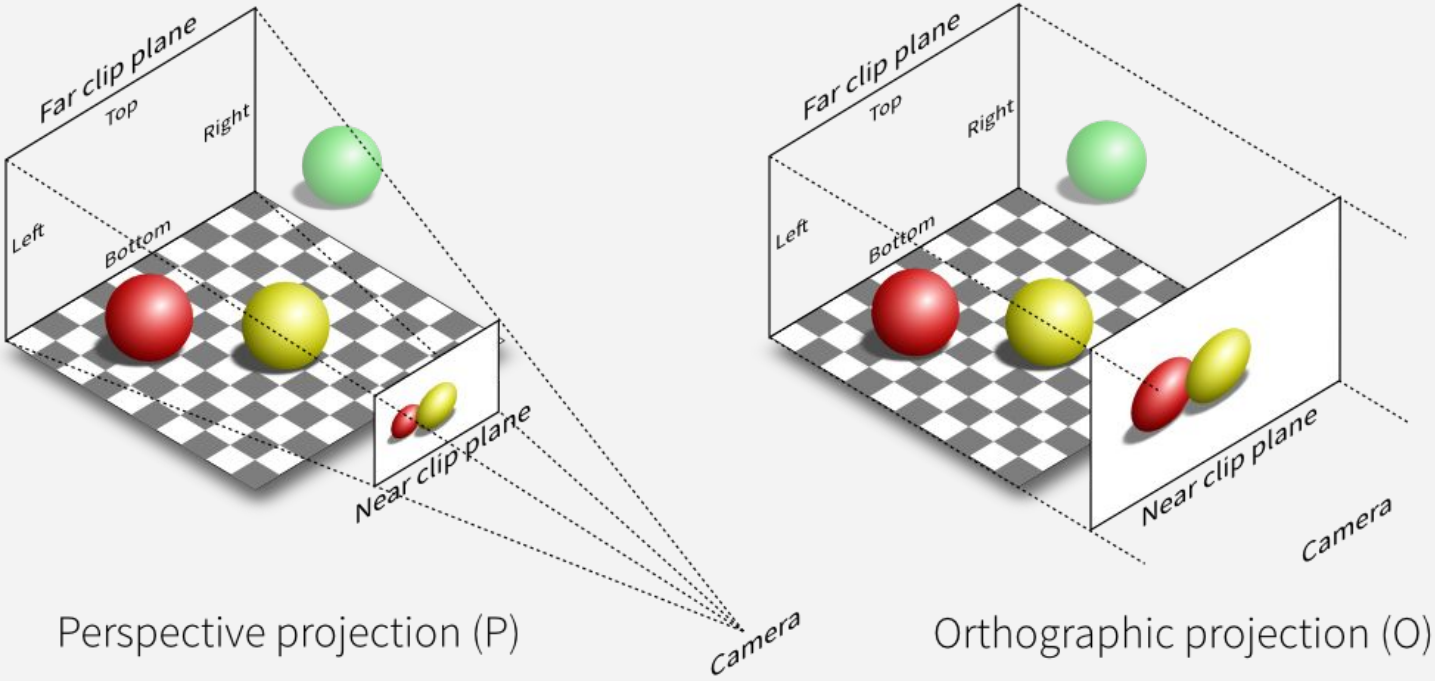
https://openai.com/blog/generative-models/

Background: Object Space -> Screen Space



https://stackoverflow.com/questions/29061257/get-pure-view-matrix-in-vuforia

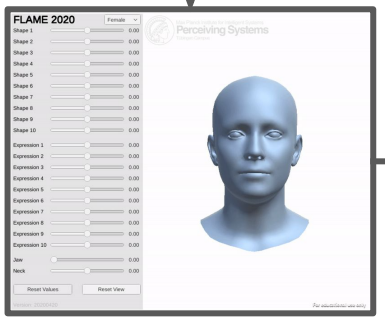
Background: Perspective Projection



https://stackoverflow.com/questions/36573283/from-perspective-picture-to-orthographic-picture
http://learnwebgl.brown37.net/08_projections/projections_perspective.html

Face2Face: Generative Model

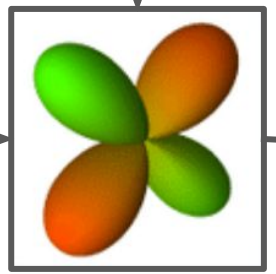
Identity (α)
Expression (δ)



Morphable Model

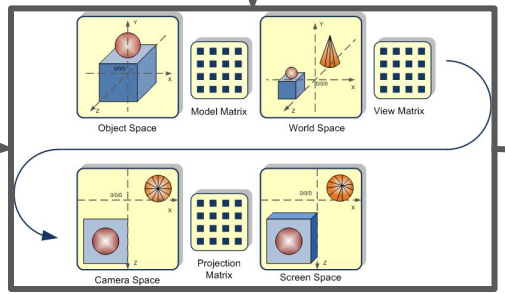
3d mesh

Appearance (β)



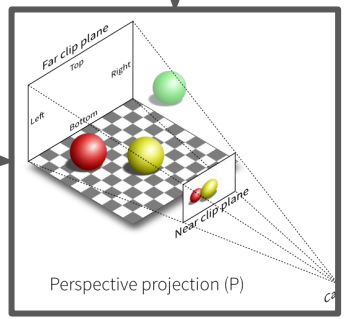
Spherical Harmonic
Illumination*

Rotation (R)
Translation (t)



Object -> Camera Space

Camera Info (k)



Perspective Projection

Output: 2D Image

Face2Face: Generative Model

A multi-linear PCA model is used, where parameter consists of facial identity (geometric shape and skin reflectance) and facial expression.

$$\begin{aligned} \mathcal{M}_{\text{geo}}(\boldsymbol{\alpha}, \boldsymbol{\delta}) &= \boldsymbol{a}_{\text{id}} + E_{\text{id}} \cdot \boldsymbol{\alpha} + E_{\text{exp}} \cdot \boldsymbol{\delta} \\ \mathcal{M}_{\text{alb}}(\boldsymbol{\beta}) &= \boldsymbol{a}_{\text{alb}} + E_{\text{alb}} \cdot \boldsymbol{\beta} \ . \end{aligned}$$

Here we estimate $\boldsymbol{\alpha}$,$\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ while rest of the parameters are knows to us. We assume estimated face to be a multivariate normal distribution around average shape $\boldsymbol{a}_{\text{id}}$ and reflectance $\boldsymbol{a}_{\text{alb}}$

Face2Face: Estimation of parameters

Given an **input 2d image**, use the following loss to estimate parameters **α**,**β** and ***δ***

$$E(\boldsymbol{\mathcal{P}}) = \underbrace{w_{col}E_{col}(\boldsymbol{\mathcal{P}}) + w_{lan}E_{lan}(\boldsymbol{\mathcal{P}})}_{data} + \underbrace{w_{reg}E_{reg}(\boldsymbol{\mathcal{P}})}_{prior}$$

Here E_{col} , E_{lan} and E_{reg} are photo consistency, feature alignment and statistical regularization terms respectively.

$$E_{\text{col}}(\boldsymbol{\mathcal{P}}) = \frac{1}{|\mathcal{V}|}\sum_{\boldsymbol{p}\in\mathcal{V}} \|C_{\mathcal{S}}(\boldsymbol{p}) - C_{\mathcal{I}}(\boldsymbol{p})\|_2$$

$$E_{\text{lan}}(\boldsymbol{\mathcal{P}}) = \frac{1}{|\mathcal{F}|}\sum_{\boldsymbol{f}_j\in\mathcal{F}} w_{\text{conf},j}\,\|\boldsymbol{f}_j - \Pi(\Phi(\boldsymbol{v}_j)\|_2^2 \ .$$

$$E_{\text{reg}}(\boldsymbol{\mathcal{P}}) = \sum_{i=1}^{80}\left[\left(\frac{\boldsymbol{\alpha}_i}{\sigma_{\text{id},i}}\right)^2 + \left(\frac{\boldsymbol{\beta}_i}{\sigma_{\text{alb},i}}\right)^2\right] + \sum_{i=1}^{76}\left(\frac{\boldsymbol{\delta}_i}{\sigma_{\text{exp},i}}\right)^2 \ .$$

Training Details: Data-Parallel Optimization

Data-Parallel Optimization Strategy

- **[GN]** Iteratively-Reweighted Least Squares
- **[GN]** Gauss-Newton
- **[PCG]** Preconditioned Conjugate-Gradient

Training Details: Non-Rigid Model-Based Bundling

Monocular reconstruction is an **under-constrained** optimization problem.

- Many parameters, and few constraints
- Optimization problem has infinitely many “correct” solutions

Face2Face uses **multiple video frames** to learn the identity parameters.

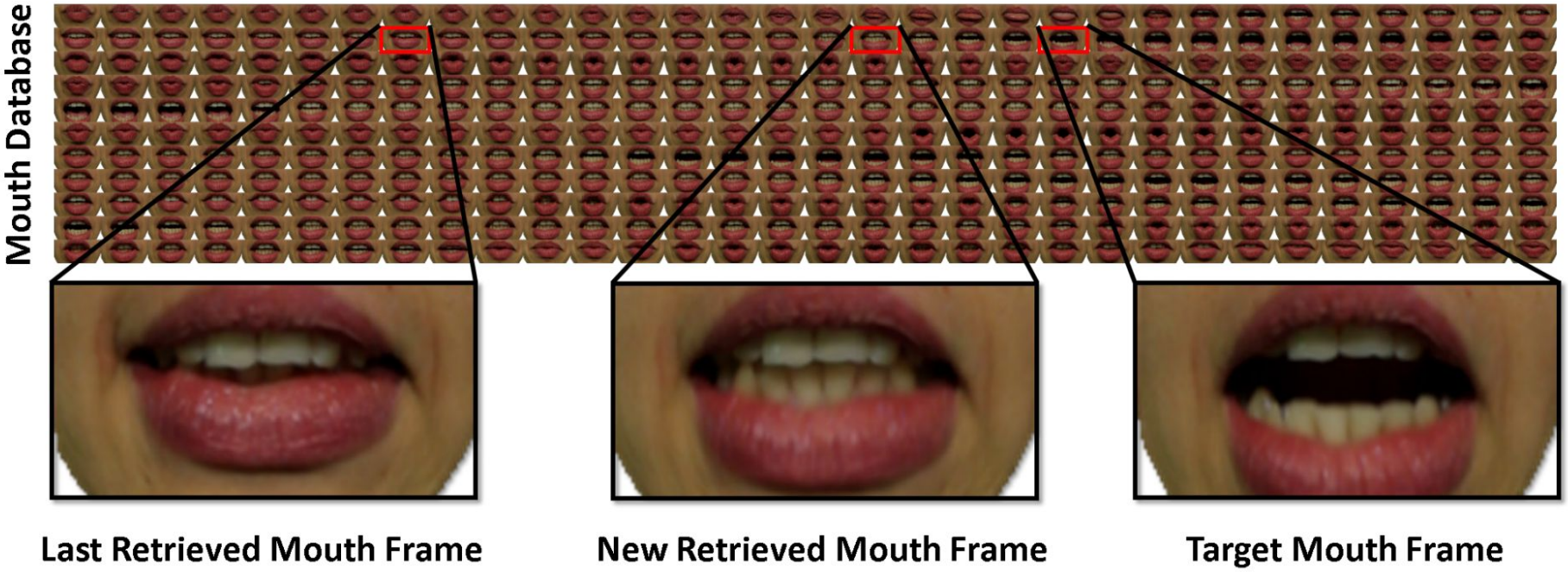
- Identity (alpha), Color (beta), and Camera (kappa) parameters never change
- Expression (delta), Illumination (gamma), Position (R,t) change every frame

Face2Face requires a short **“warm-up” period** to learn the “identity” parameters before real-time re-enactment can begin.

Mouth Interior Synthesis (Older Methods…)

Older methods copied the source mouth onto the target image, or used a 3d model:

Mouth Interior Synthesis (3/N)



Mouth Retrieval: we use an appearance graph to retrieve new mouth frames. In order to select a frame, we enforce similarity to the previously-retrieved frame while minimizing the distance to the target expression.

MoFA

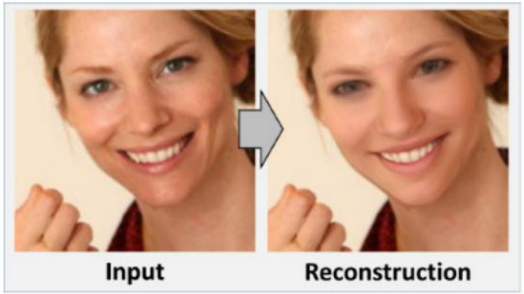
[Tewari 2017] “MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction”

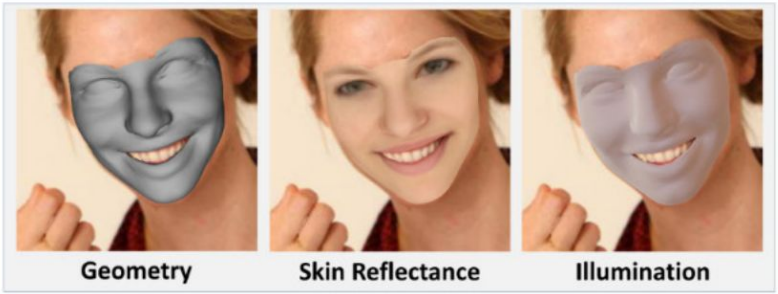
MoFA: Overview

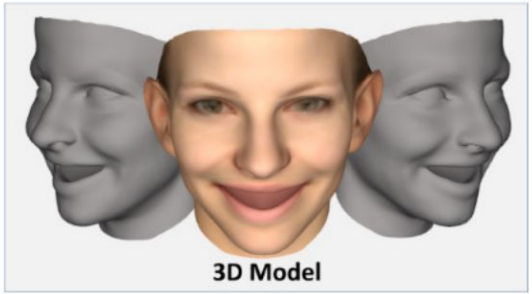
Like Face2Face, the goal of MoFA is **real-time monocular reconstruction**.

- Combines “generative” methods and “regression” methods
- Compared to optimization-based approaches, MoFA efficiently **regresses** model parameters **without iterative optimization**

Innovation is to have an end to end model which can be trained unsupervised manner







Related: Monocular Optimization-based Reconstruction

The basis vectors and parameters to be estimated are same as that of Face2Face. This are also called semantic vector which consists shape, expression and variation

Here a encoder is used which taken in the image and outputs semantic vector which is then further used by a decoder which outputs the generated image with landmark points.

Loss function used here is same as that used in Face2Face.

$$E(\boldsymbol{\mathcal{P}}) = \underbrace{w_{col}E_{col}(\boldsymbol{\mathcal{P}}) + w_{lan}E_{lan}(\boldsymbol{\mathcal{P}})}_{data} + \underbrace{w_{reg}E_{reg}(\boldsymbol{\mathcal{P}})}_{prior}$$

Background: Autoencoders

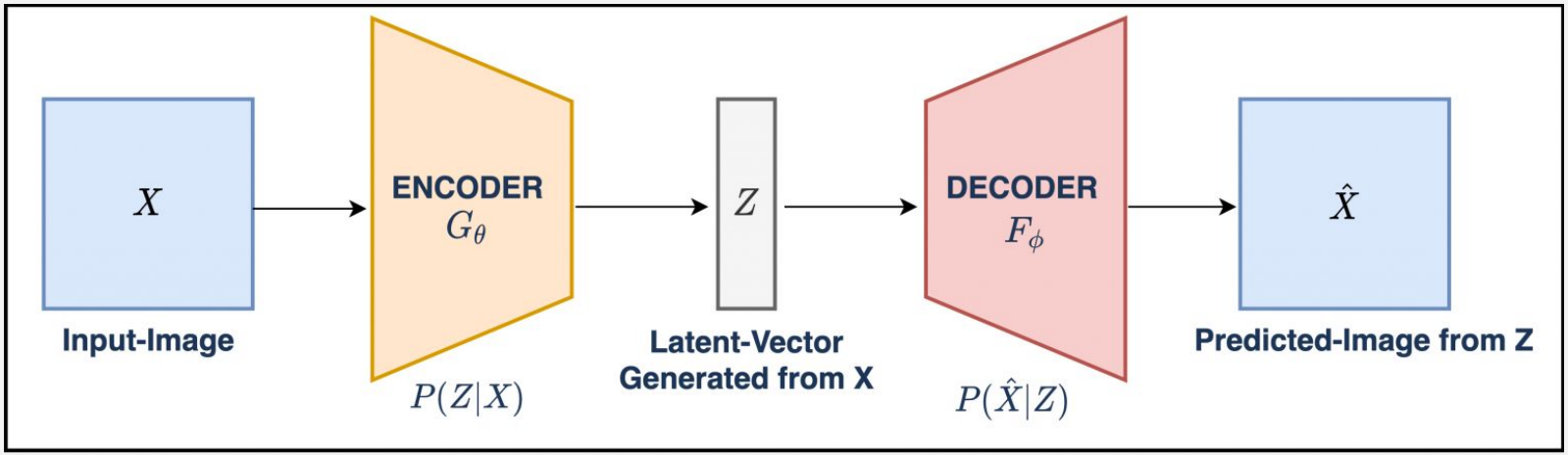


Image: https://learnopencv.com/autoencoder-in-tensorflow-2-beginners-guide/

Background: Autoencoder vs PCA

Main difference between Autoencoder and PCA is that autoencoder is capable of modelling (or compressing) complex non linear data (thanks for NN and non linearity) much better.

This added functionality comes with a price, which is computational complexity. Because of linear nature of PCA, its computational complexity is way less than that of autoencoder.

Linear Autoencoders learn the same latent subspace as PCA:

- Plaut 2018, “From Principal Subspaces to Principal Components with Linear Autoencoders”

MoFA Architecture

(decoder can use any 3dmm)

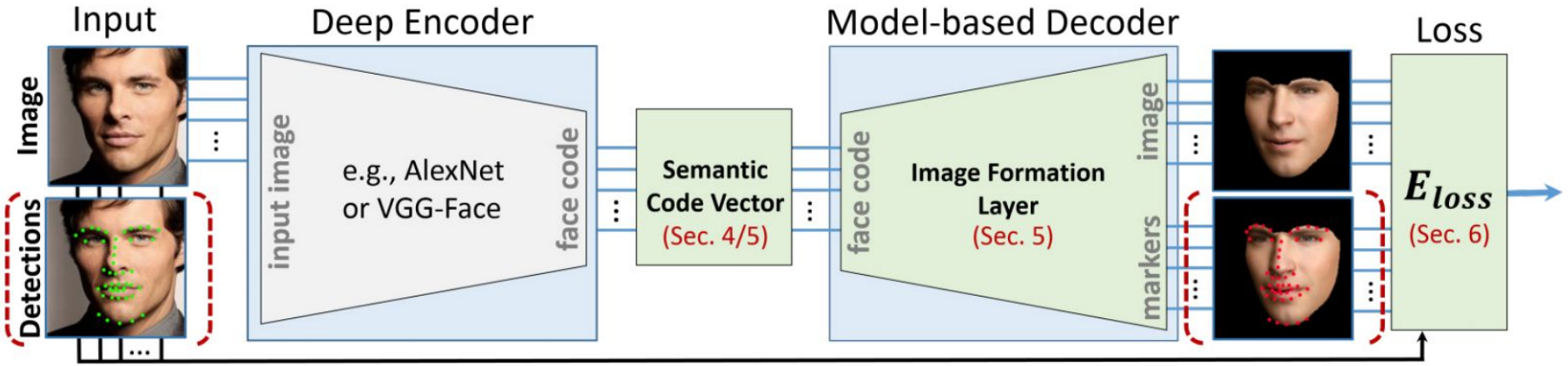


Figure 1. Our deep model-based face autoencoder enables unsupervised end-to-end learning of semantic parameters, such as pose, shape, expression, skin reflectance and illumination. An optional landmark-based surrogate loss enables faster convergence and improved reconstruction results, see Sec. 6. Both scenarios require no supervision of the semantic parameters during training.

(MoFa) Semantic Code Vector

MoFA uses the same latent semantic code vector as Face2Face:

Semantic code vector $x = (\alpha, \beta, \delta, T, t, \gamma) \in \mathbb{R}^{257}$

- facial expression $\delta \in \mathbb{R}^{64}$
- shape $\alpha \in \mathbb{R}^{80}$
- skin reflectance $\beta \in \mathbb{R}^{80}$
- camera rotation $T \in SO(3)$
- translation $t \in \mathbb{R}^3$
- scene illumination $\gamma \in \mathbb{R}^{27}$

MoFA: Loss Functions

$$E_{\text{loss}}(\mathbf{x}) = \underbrace{w_{\text{land}} E_{\text{land}}(\mathbf{x}) + w_{\text{photo}} E_{\text{photo}}(\mathbf{x})}_{\text{data term}} + \underbrace{w_{\text{reg}} E_{\text{reg}}(\mathbf{x})}_{\text{regularizer}} \quad .$$

Dense Photometric Alignment:

Sparse Landmark Alignment: L2 Loss on detected facial landmarks between source/target images.

Regularizer:

Face2Face

MoFA